CHAPTER 4
Number Theory and Cryptography

SECTION 4.1 Divisibility and Modular Arithmetic

Number theory is playing an increasingly important role in computer science. This section and these exercises just scratch the surface of what is relevant. Many of these exercises are simply a matter of applying definitions. It is sometimes hard for a beginning student to remember that in order to prove something about a concept (such as modular arithmetic), it is usually necessary to invoke the definition! Exercises 34–44 hint at the rich structure that modular arithmetic has (sometimes resembling real number arithmetic more than integer arithmetic). In many contexts in mathematics and computer science, modular arithmetic is more relevant and convenient than ordinary integer arithmetic.

1. **a)** yes, since $68 = 17 \cdot 4$ **b)** no, remainder $= 16$
c) yes, since $357 = 17 \cdot 21$ **d)** no, remainder $= 15$

3. If $a \mid b$, then we know that $b = at$ for some integer t. Therefore $bc = a(tc)$, so by definition $a \mid bc$.

5. The given conditions imply that there are integers s and t such that $a = bs$ and $b = at$. Combining these, we obtain $a = ats$; since $a \neq 0$, we conclude that $st = 1$. Now the only way for this to happen is for $s = t = 1$ or $s = t = -1$. Therefore either $a = b$ or $a = -b$.

7. The given condition means that $bc = (ac)t$ for some integer t. Since $c \neq 0$, we can divide both sides by c to obtain $b = at$. This is the definition of $a \mid b$, as desired.

9. In each case we need to find (the unique integers) q and r such that $a = dq + r$ and $0 \leq r < d$, where a and d are the given integers. In each case $q = \lfloor a/d \rfloor$.
a) $19 = 7 \cdot 2 + 5$, so $q = 2$ and $r = 5$ **b)** $-111 = 11 \cdot (-11) + 10$, so $q = -11$ and $r = 10$
c) $789 = 23 \cdot 34 + 7$, so $q = 34$ and $r = 7$ **d)** $1001 = 13 \cdot 77 + 0$, so $q = 77$ and $r = 0$
e) $0 = 19 \cdot 0 + 0$, so $q = 0$ and $r = 0$ **f)** $3 = 5 \cdot 0 + 3$, so $q = 0$ and $r = 3$
g) $-1 = 3 \cdot (-1) + 2$, so $q = -1$ and $r = 2$ **h)** $4 = 1 \cdot 4 + 0$, so $q = 4$ and $r = 0$

11. We are doing arithmetic modulo 12 for this exercise.
a) Because $11 + 80 \textbf{ mod } 12 = 7$, the clock reads 7:00.
b) Because $12 - 40 \textbf{ mod } 12 = -28 \textbf{ mod } 12 = -28 + 36 \textbf{ mod } 12 = 8$, the clock reads 8:00.
c) Because $6 + 100 \textbf{ mod } 12 = 10$, the clock reads 10:00.

13. In each case we merely have to compute the expression on the right **mod** 13. This means dividing it by 13 and taking the (nonnegative) remainder.
a) $9 \cdot 4 \textbf{ mod } 13 = 36 \textbf{ mod } 13 = 10$ **b)** $11 \cdot 9 \textbf{ mod } 13 = 99 \textbf{ mod } 13 = 8$
c) $4 + 9 \textbf{ mod } 13 = 13 \textbf{ mod } 13 = 0$ **d)** $2 \cdot 4 + 3 \cdot 9 \textbf{ mod } 13 = 35 \textbf{ mod } 13 = 9$
e) $4^2 + 9^2 \textbf{ mod } 13 = 97 \textbf{ mod } 13 = 6$
f) $4^3 - 9^3 \textbf{ mod } 13 = -665 \textbf{ mod } 13 = 11$ (because $-665 = -52 \cdot 13 + 11$)

15. The given condition, that a **mod** $m = b$ **mod** m, means that a and b have the same remainder when divided by m. In symbols, $a = q_1 m + r$ and $b = q_2 m + r$ for some integers q_1, q_2, and r. Subtracting these two equations gives us $a - b = (q_1 - q_2)m$, which says that m divides (is a factor of) $a - b$. This is precisely the definition of $a \equiv b \pmod{m}$.

17. The quotient n/k lies between two consecutive integers, say $b-1$ and b, possibly equal to b. In symbols, there exists a positive integer b such that $b - 1 < n/k \leq b$. In particular, $\lceil n/k \rceil = b$. Also, since $n/k > b - 1$, we have $n > k(b-1)$, and so (since everything is an integer) $n - 1 \geq k(b-1)$. This means that $(n-1)/k \geq b-1$, so $\lfloor (n-1)/k \rfloor \geq b-1$. On the other hand, $\lfloor (n-1)/k \rfloor \leq (n-1)/k < n/k \leq b$, so $\lfloor (n-1)/k \rfloor < b$. Therefore $\lfloor (n-1)/k \rfloor = b - 1$. The desired conclusion follows.

19. Let's first look at an example or two. If $m = 7$, then the usual set of values we use for the congruence classes modulo m is $\{0, 1, 2, 3, 4, 5, 6\}$. However, we can replace 6 by -1, 5 by -2, and 4 by -3 to get the collection $\{-3, -2, -1, 0, 1, 2, 3\}$. These will be the values with smallest absolute values. Similarly, if $m = 8$, then the collection we want is $\{-3, -2, -1, 0, 1, 2, 3, 4\}$ ($\{-4, -3, -2, -1, 0, 1, 2, 3\}$ would do just as well). In general, in place of $\{0, 1, 2, \ldots, m-1\}$ we can use $\{\lceil -m/2 \rceil, \lceil -m/2 \rceil + 1, \ldots, -1, 0, 1, 2, \ldots, \lceil m/2 \rceil\}$, omitting either $\lceil -m/2 \rceil$ or $\lceil m/2 \rceil$ if m is even. Note that the values in $\{0, 1, 2, \ldots, m-1\}$ greater than $\lceil m/2 \rceil$ have had m subtracted from them to produce the negative values in our answer. As for a formula to produce these values, we can use a two-part formula:

$$f(x) = \begin{cases} x \textbf{ mod } m & \text{if } x \textbf{ mod } m \leq \lceil m/2 \rceil \\ (x \textbf{ mod } m) - m & \text{if } x \textbf{ mod } m > \lceil m/2 \rceil. \end{cases}$$

Note that if m is even, then we can, alternatively, take $f(m/2) = -m/2$.

21. For these problems, we need to perform the division (as in Exercise 9) and report the remainder.

a) $13 = 3 \cdot 4 + 1$, so 13 **mod** $3 = 1$ **b)** $-97 = 11 \cdot (-9) + 2$, so -97 **mod** $11 = 2$

c) $155 = 19 \cdot 8 + 3$, so 155 **mod** $19 = 3$ **d)** $-221 = 23 \cdot (-10) + 9$, so -221 **mod** $23 = 9$

23. Recall that a **div** m and a **mod** m are the integer quotient and remainder when a is divided by m.

a) Because $228 = 1 \cdot 119 + 109$, we have 228 **div** $119 = 1$ and 228 **mod** $119 = 109$.

b) Because $9009 = 40 \cdot 223 + 89$, we have 9009 **div** $223 = 40$ and 9009 **mod** $223 = 89$.

c) Because $-10101 = -31 \cdot 333 + 222$, we have -10101 **div** $333 = -31$ and -10101 **mod** $333 = 222$. (Note that $10101 \div 333$ is $30\frac{111}{333}$, so without the negative dividend we would get a different absolute quotient and different remainder. But we have to round the negative quotient here, $-30\frac{111}{333}$, *down* to -31 in order for the remainder to be nonnegative.)

d) Because $-765432 = -21 \cdot 38271 + 38259$, we have -765432 **div** $38271 = -21$ and -765432 **mod** $38271 = 38259$.

25. **a)** Because -15 already satisfies the inequality, the answer is -15.

b) Because 24 is too large to satisfy the inequality, we subtract 31 and obtain the answer is -7.

c) Because 99 is too small to satisfy the inequality, we add 41 and obtain the answer is 140.

27. We just need to start at -1 and repeatedly subtract or add 25 until we exceed the desired range. Thus the negative values we seek are -1, -26, -51, and -76, and the positive values are 24, 49, 74, and 99.

29. For these problems, we need to divide by 17 and see whether the remainder equals 5. Remember that the quotient can be negative, but the remainder r must satisfy $0 \leq r < 17$.

a) $80 = 17 \cdot 4 + 12$, so $80 \not\equiv 5 \pmod{17}$ **b)** $103 = 17 \cdot 6 + 1$, so $103 \not\equiv 5 \pmod{17}$

c) $-29 = 17 \cdot (-2) + 5$, so $-29 \equiv 5 \pmod{17}$ **d)** $-122 = 17 \cdot (-8) + 14$, so $-122 \not\equiv 5 \pmod{17}$

31.

a) Working modulo 23, we have $-133 + 261 = 128 \equiv 13$, so the answer is 13.

b) Working modulo 23, we have $457 \cdot 182 \equiv 20 \cdot 21 = 420 \equiv 6$.

33. **a)** $(99^2 \textbf{ mod } 32)^3 \textbf{ mod } 15 = (3^2 \textbf{ mod } 32)^3 \textbf{ mod } 15 = 9^3 \textbf{ mod } 15 = 729 \textbf{ mod } 15 = 9$

b) $(3^4 \textbf{ mod } 17)^2 \textbf{ mod } 11 = (81 \textbf{ mod } 17)^2 \textbf{ mod } 11 = 13^2 \textbf{ mod } 11 = 2^2 \textbf{ mod } 11 = 4$

c) $(19^3 \textbf{ mod } 23)^2 \textbf{ mod } 31 = ((-4)^3 \textbf{ mod } 23)^2 \textbf{ mod } 31 = (-64 \textbf{ mod } 23)^2 \textbf{ mod } 31 = 5^2 \textbf{ mod } 31 = 25$

d) $(89^3 \textbf{ mod } 79)^4 \textbf{ mod } 26 = (10^3 \textbf{ mod } 79)^4 \textbf{ mod } 26 = (1000 \textbf{ mod } 79)^4 \textbf{ mod } 26 = 52^4 \textbf{ mod } 26 = 0^4 \textbf{ mod } 26 = 0$

35. The hypothesis $a \equiv b \pmod{m}$ means that $m \,|\, (a - b)$. Since we are given that $n \,|\, m$, Theorem 1(*iii*) implies that $n \,|\, (a - b)$. Therefore $a \equiv b \pmod{n}$, as desired.

37. **a)** To show that this conditional statement does not necessarily hold, we need to find an example in which $ac \equiv bc \pmod{m}$, but $a \not\equiv b \pmod{m}$. Let $m = 4$ and $c = 2$ (what is important in constructing this example is that m and c have a nontrivial common factor). Let $a = 0$ and $b = 2$. Then $ac = 0$ and $bc = 4$, so $ac \equiv bc \pmod{4}$, but $0 \not\equiv 2 \pmod{4}$.

b) To show that this conditional statement does not necessarily hold, we need to find an example in which $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, but $a^c \not\equiv b^d \pmod{m}$. If we try a few randomly chosen positive integers, we will soon find one. Let $m = 5$, $a = 3$, $b = 3$, $c = 1$, and $d = 6$. Then $a^c = 3$ and $b^d = 729 \equiv 4 \pmod{5}$, so $3^1 \not\equiv 3^6 \pmod{5}$, even though $3 \equiv 3 \pmod{5}$ and $1 \equiv 6 \pmod{5}$.

39. By Exercise 38 the sum of two squares must be either $0 + 0$, $0 + 1$, or $1 + 1$, modulo 4. Therefore the sum cannot be 3 modulo 4, which means that it cannot be of the form $4k + 3$.

41. There are at least two ways to prove this. One way is to invoke Theorem 5 repeatedly. Since $a \equiv b \pmod{m}$, Theorem 5 implies that $a \cdot a \equiv b \cdot b \pmod{m}$, i.e., $a^2 \equiv b^2 \pmod{m}$. Invoking Theorem 5 again, since $a \equiv b \pmod{m}$ and $a^2 \equiv b^2 \pmod{m}$, we obtain $a^3 \equiv b^3 \pmod{m}$. After $k - 1$ applications of this process, we obtain $a^k \equiv b^k \pmod{m}$, as desired. (This is really a proof by mathematical induction, a topic to be considered formally in Chapter 5.)

Alternately, we can argue directly, using the algebraic identity $a^k - b^k = (a-b)(a^{k-1} + a^{k-2}b + \cdots + ab^{k-2} + b^{k-1})$. Specifically, the hypothesis that $a \equiv b \pmod{m}$ means that $m \,|\, (a - b)$. Therefore by Theorem 1(*ii*), m divides the right-hand side of this identity, so $m \,|\, (a^k - b^k)$. This means precisely that $a^k \equiv b^k \pmod{m}$.

43. The closure property states that $a \cdot_m b \in \mathbf{Z}_m$ whenever $a, b \in \mathbf{Z}_m$. Recall that $\mathbf{Z}_m = \{0, 1, 2, \ldots, m - 1\}$ and that $a \cdot_m b$ is defined to be $(a \cdot b) \textbf{ mod } m$. But this last expression will by definition be an integer in the desired range. To see that multiplication is associative, we must show that $(a \cdot_m b) \cdot_m c = a \cdot_m (b \cdot_m c)$. This is equivalent to

$$((a \cdot b \textbf{ mod } m) \cdot c) \textbf{ mod } m = (a \cdot (b \cdot c \textbf{ mod } m)) \textbf{ mod } m.$$

This is true, because both sides equal $(a \cdot b \cdot c) \textbf{ mod } m$ (multiplication of integers is associative). Similarly, multiplication in $\mathbf{Z}_m$ is commutative because multiplication in $\mathbf{Z}$ is commutative, and 1 is the multiplicative identity for $\mathbf{Z}_m$ because 1 is the multiplicative identity for $\mathbf{Z}$.

45. We will use $+$ and $\cdot$ for these operations to save space and improve the appearance of the table. Notice that we really can get by with a little more than half of this table if we observe that these operations are commutative; then we would need to list $a + b$ and $a \cdot b$ only for $a \leq b$.

$$
\begin{array}{lllll}
0+0=0 & 0+1=1 & 0+2=2 & 0+3=3 & 0+4=4 \\
1+0=1 & 1+1=2 & 1+2=3 & 1+3=4 & 1+4=0 \\
2+0=2 & 2+1=3 & 2+2=4 & 2+3=0 & 2+4=1 \\
3+0=3 & 3+1=4 & 3+2=0 & 3+3=1 & 3+4=2 \\
4+0=4 & 4+1=0 & 4+2=1 & 4+3=2 & 4+4=3
\end{array}
$$

$$
\begin{array}{lllll}
0\cdot 0=0 & 0\cdot 1=0 & 0\cdot 2=0 & 0\cdot 3=0 & 0\cdot 4=0 \\
1\cdot 0=0 & 1\cdot 1=1 & 1\cdot 2=2 & 1\cdot 3=3 & 1\cdot 4=4 \\
2\cdot 0=0 & 2\cdot 1=2 & 2\cdot 2=4 & 2\cdot 3=1 & 2\cdot 4=3 \\
3\cdot 0=0 & 3\cdot 1=3 & 3\cdot 2=1 & 3\cdot 3=4 & 3\cdot 4=2 \\
4\cdot 0=0 & 4\cdot 1=4 & 4\cdot 2=3 & 4\cdot 3=2 & 4\cdot 4=1
\end{array}
$$

47. If $d = 1$, then $f(a) = a$ and $g(a) = 0$. Therefore f is clearly one-to-one and onto, and g is neither. If $d > 1$, then f is still onto, because $f(db) = b$ for any desired $b \in \mathbf{Z}$, but it is clearly not one-to-one, because $f(0) = f(1) = 0$. Furthermore, g is clearly not onto, because its range is just $\{0, 1, 2, \ldots, d-1\}$, and it is not one-to-one because $g(0) = g(d) = 0$.

SECTION 4.2 Integer Representations and Algorithms

In addition to having some routine calculation exercises, this exercise set introduces other forms of representing integers. These are **balanced ternary expansion**, **Cantor expansion**, **binary coded decimal** *(or* **BCD***) representation, and* **one's** *and* **two's complement** *representations. Each has practical and/or theoretical importance in mathematics or computer science. If all else fails, one can carry out an algorithm by "playing computer" and mechanically following the pseudocode step by step.*

1. We divide repeatedly by 2, noting the remainders. The remainders are then arranged from right to left to obtain the binary representation of the given number.

a) We begin by dividing 231 by 2, obtaining a quotient of 115 and a remainder of 1. Therefore $a_0 = 1$. Next $115/2 = 57$, remainder 1. Therefore $a_1 = 1$. Similarly $57/2 = 28$, remainder 1. Therefore $a_2 = 1$. Then $28/2 = 14$, remainder 0, so $a_3 = 0$. Similarly $a_4 = 0$, after we divide 14 by 2, obtaining 7 with remainder 0. Three more divisions yield quotients of 3, 1, and 0, with remainders of 1, 1, and 1, respectively, so $a_5 = a_6 = a_7 = 1$. Putting all this together, we see that the binary representation is $(a_7a_6a_5a_4a_3a_2a_1a_0)_2 = (1110\ 0111)_2$. As a check we can compute that $2^0 + 2^1 + 2^2 + 2^5 + 2^6 + 2^7 = 231$.

b) Following the same procedure as in part **(a)**, we obtain successive remainders 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1. Therefore $4532 = (1\ 0001\ 1011\ 0100)_2$.

c) By the same method we obtain $97644 = (1\ 0111\ 1101\ 0110\ 1100)_2$.

3. **a)** $(1\ 1111)_2 = 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 16 + 8 + 4 + 2 + 1 = 31$. An easier way to get the answer is to note that $(1\ 1111)_2 = (10\ 0000)_2 - 1 = 2^5 - 1 = 31$.

b) $(10\ 0000\ 0001)_2 = 2^9 + 2^0 = 513$

c) $(1\ 0101\ 0101)_2 = 2^8 + 2^6 + 2^4 + 2^2 + 2^0 = 256 + 64 + 16 + 4 + 1 = 341$

d) $(110\ 1001\ 0001\ 0000)_2 = 2^{14} + 2^{13} + 2^{11} + 2^8 + 2^4 = 16384 + 8192 + 2048 + 256 + 16 = 26896$

5. In each case we follow the idea given in Example 7, converting each octal digit to its binary equivalent (including leading 0's where necessary). Note that by convention we then group the binary digits into groups of fours, starting at the right.

a) Since $(5)_8 = (101)_2$, $(7)_8 = (111)_2$, and $(2)_8 = (010)_2$, we have $(572)_8 = (1\ 0111\ 1010)_2$.

b) We concatenate 1, 110, 000, and 100 to obtain $(11\ 1000\ 0100)_2$.

c) $(1\ 0001\ 0011)_2$ **d)** $(101\ 0000\ 1111)_2$

7. Following Example 7, we simply write the binary equivalents of each digit: $(\text{A})_{16} = (1010)_2$, $(\text{B})_{16} = (1011)_2$, $(\text{C})_{16} = (1100)_2$, $(\text{D})_{16} = (1101)_2$, $(\text{E})_{16} = (1110)_2$, and $(\text{F})_{16} = (1111)_2$. Note that the blocking by groups of four binary digits is just for readability by humans.

a) $(80\text{E})_{16} = (1000\ 0000\ 1110)_2$

b) $(135\text{AB})_{16} = (0001\ 0011\ 0101\ 1010\ 1011)_2$

c) $(\text{ABBA})_{16} = (1010\ 1011\ 1011\ 1010)_2$

d) $(\text{DEFACED})_{16} = (1101\ 1110\ 1111\ 1010\ 1100\ 1110\ 1101)_2$

9. Following Example 7, we simply write the binary equivalents of each digit. Since $(\text{A})_{16} = (1010)_2$, $(\text{B})_{16} = (1011)_2$, $(\text{C})_{16} = (1100)_2$, $(\text{D})_{16} = (1101)_2$, $(\text{E})_{16} = (1110)_2$, and $(\text{F})_{16} = (1111)_2$, we see that $(\text{ABCDEF})_{16} = (101010111100110111101111)_2$. Following the convention shown in Exercise 3 of grouping binary digits by fours, we can write this in a more readable form as 1010 1011 1100 1101 1110 1111.

11. Following Example 7, we simply write the hexadecimal equivalents of each group of four binary digits. Thus we have $(1011\ 0111\ 1011)_2 = (\text{B7B})_{16}$.

13. We adopt a notation that will help with the explanation. Adding up to three leading 0's if necessary, write the binary expansion as $(\ldots b_{23}b_{22}b_{21}b_{20}b_{13}b_{12}b_{11}b_{10}b_{03}b_{02}b_{01}b_{00})_2$. The value of this numeral is $b_{00} + 2b_{01} + 4b_{02} + 8b_{03} + 2^4b_{10} + 2^5b_{11} + 2^6b_{12} + 2^7b_{13} + 2^8b_{20} + 2^9b_{21} + 2^{10}b_{22} + 2^{11}b_{23} + \cdots$, which we can rewrite as $b_{00} + 2b_{01} + 4b_{02} + 8b_{03} + (b_{10} + 2b_{11} + 4b_{12} + 8b_{13}) \cdot 2^4 + (b_{20} + 2b_{21} + 4b_{22} + 8b_{23}) \cdot 2^8 + \cdots$. Now $(b_{i3}b_{i2}b_{i1}b_{i0})_2$ translates into the hexadecimal digit h_i. So our number is $h_0 + h_1 \cdot 2^4 + h_2 \cdot 2^8 + \cdots = h_0 + h_1 \cdot 16 + h_2 \cdot 16^2 + \cdots$, which is the hexadecimal expansion $(\ldots h_1h_1h_0)_{16}$.

15. We adopt a notation that will help with the explanation. Adding up to two leading 0's if necessary, write the binary expansion as $(\ldots b_{22}b_{21}b_{20}b_{12}b_{11}b_{10}b_{02}b_{01}b_{00})_2$. The value of this numeral is $b_{00} + 2b_{01} + 4b_{02} + 2^3b_{10} + 2^4b_{11} + 2^5b_{12} + 2^6b_{20} + 2^7b_{21} + 2^8b_{22} + \cdots$, which we can rewrite as $b_{00} + 2b_{01} + 4b_{02} + (b_{10} + 2b_{11} + 4b_{12}) \cdot 2^3 + (b_{20} + 2b_{21} + 4b_{22}) \cdot 2^6 + \cdots$. Now $(b_{i2}b_{i1}b_{i0})_2$ translates into the octal digit h_i. So our number is $h_0 + h_1 \cdot 2^3 + h_2 \cdot 2^6 + \cdots = h_0 + h_1 \cdot 8 + h_2 \cdot 8^2 + \cdots$, which is the octal expansion $(\ldots h_1h_1h_0)_8$.

17. In each case we follow the method of Example 7, blocking by threes instead of fours. We replace each octal digit of the given numeral by its 3-digit binary equivalent and string the digits together. The first digit is $(7)_8 = (111)_2$, the next is $(3)_8 = (011)_2$, and so on, so we obtain $(1\ 1101\ 1100\ 1010\ 1101\ 0001)_2$. For the other direction, we split the given binary numeral into blocks of three digits, adding initial 0's to fill it out: 001 010 111 011. Then we replace each block by its octal equivalent, obtaining the answer $(1273)_8$.

19. Since we have procedures for converting both octal and hexadecimal to and from binary (Example 7), to convert from octal to hexadecimal, we first convert from octal to binary and then convert from binary to hexadecimal.

21. We can just add and multiply using the grade-school algorithms, working with these very simple addition and multiplication tables: $0 + 0 = 0$, $0 + 1 = 1 + 0 = 1$, $1 + 1 = 10$, which means that we "carry" the 1 into the

next column; $0 \cdot 0 = 0 \cdot 1 = 1 \cdot 0 = 0$, $1 \cdot 1 = 1$. See Examples 8 and 10. Note that we can check our work by converting everything to decimal numerals (the check is shown in parentheses below). For convenience, we leave off the "2" subscripts throughout.

a) $100\ 0111 + 111\ 0111 = 1011\ 1110$ (decimal: $71 + 119 = 190$)
$100\ 0111 \cdot 111\ 0111 = 10\ 0001\ 0000\ 0001$ (decimal: $71 \cdot 119 = 8449$)

b) $1110\ 1111 + 1011\ 1101 = 1\ 1010\ 1100$ (decimal: $239 + 189 = 428$)
$1110\ 1111 \cdot 1011\ 1101 = 1011\ 0000\ 0111\ 0011$ (decimal: $239 \cdot 189 = 45{,}171$)

c) $10\ 1010\ 1010 + 1\ 1111\ 0000 = 100\ 1001\ 1010$ (decimal: $682 + 496 = 1178$)
$10\ 1010\ 1010 \cdot 1\ 1111\ 0000 = 101\ 0010\ 1001\ 0110\ 0000$ (decimal: $682 \cdot 496 = 338{,}272$)

d) $10\ 0000\ 0001 + 11\ 1111\ 1111 = 110\ 0000\ 0000$ (decimal: $513 + 1023 = 1536$)
$10\ 0000\ 0001 \cdot 11\ 1111\ 1111 = 1000\ 0000\ 0001\ 1111\ 1111$ (decimal: $513 \cdot 1023 = 524{,}799$)

23. We can just add and multiply using the grade-school algorithms (working column by column starting at the right), using the addition and multiplication tables in base eight (for example, $5 + 6 = 13$ and $5 \cdot 6 = 36$). When a digit-by-digit answer is too large to fit (i.e., greater than 7), we "carry" into the next column. Note that we can check our work by converting everything to decimal numerals (the check is shown in parentheses below). For convenience, we leave off the "8" subscripts throughout.

a) $763 + 147 = 1132$ (decimal: $499 + 103 = 602$)
$763 \cdot 147 = 144{,}305$ (decimal: $499 \cdot 103 = 51{,}397$)

b) $6001 + 272 = 6273$ (decimal: $3073 + 186 = 3259$)
$6001 \cdot 272 = 2{,}134{,}272$ (decimal: $3073 \cdot 186 = 571{,}578$)

c) $1111 + 777 = 2110$ (decimal: $585 + 511 = 1096$)
$1111 \cdot 777 = 1{,}107{,}667$ (decimal: $585 \cdot 511 = 298{,}935$)

d) $54321 + 3456 = 57{,}777$ (decimal: $22{,}737 + 1838 = 24{,}575$)
$54321 \cdot 3456 = 237{,}326{,}216$ (decimal: $22{,}737 \cdot 1838 = 41{,}790{,}606$)

25. In effect, this algorithm computes 7 **mod** 645, 7^2 **mod** 645, 7^4 **mod** 645, 7^8 **mod** 645, 7^{16} **mod** 645, ..., and then multiplies (modulo 645) the required values. Since $644 = (1010000100)_2$, we need to multiply together 7^4 **mod** 645, 7^{128} **mod** 645, and 7^{512} **mod** 645, reducing modulo 645 at each step. We compute by repeatedly squaring: 7^2 **mod** $645 = 49$, 7^4 **mod** $645 = 49^2$ **mod** $645 = 2401$ **mod** $645 = 466$, 7^8 **mod** $645 = 466^2$ **mod** $645 = 217156$ **mod** $645 = 436$, 7^{16} **mod** $645 = 436^2$ **mod** $645 = 190096$ **mod** $645 = 466$. At this point we see a pattern with period 2, so we have 7^{32} **mod** $645 = 436$, 7^{64} **mod** $645 = 466$, 7^{128} **mod** $645 = 436$, 7^{256} **mod** $645 = 466$, and 7^{512} **mod** $645 = 436$. Thus our final answer will be the product of 466, 436, and 436, reduced modulo 645. We compute these one at a time: $466 \cdot 436$ **mod** $645 = 203176$ **mod** $645 = 1$, and $1 \cdot 436$ **mod** $645 = 436$. So 7^{644} **mod** $645 = 436$. A computer algebra system will verify this; use the command "`7 &^ 644 mod 645;`" in *Maple*, for example. The ampersand here tells *Maple* to use modular exponentiation, rather than first computing the integer 7^{644}, which has over 500 digits, although it could certainly handle this if asked. The point is that modular exponentiation is much faster and avoids having to deal with such large numbers.

27. In effect, this algorithm computes 3 **mod** 99, 3^2 **mod** 99, 3^4 **mod** 99, 3^8 **mod** 99, 3^{16} **mod** 99, ..., and then multiplies (modulo 99) the required values. Since $2003 = (11111010011)_2$, we need to multiply together 3 **mod** 99, 3^2 **mod** 99, 3^{16} **mod** 99, 3^{64} **mod** 99, 3^{128} **mod** 99, 3^{256} **mod** 99, 3^{512} **mod** 99, and 3^{1024} **mod** 99, reducing modulo 99 at each step. We compute by repeatedly squaring: 3^2 **mod** $99 = 9$, 3^4 **mod** $99 = 81$, 3^8 **mod** $99 = 81^2$ **mod** $99 = 6561$ **mod** $99 = 27$, 3^{16} **mod** $99 = 27^2$ **mod** $99 = 729$ **mod** $99 = 36$, 3^{32} **mod** $99 = 36^2$ **mod** $99 = 1296$ **mod** $99 = 9$, and then the pattern repeats, so 3^{64} **mod** $99 = 81$, 3^{128} **mod** $99 = 27$, 3^{256} **mod** $99 = 36$, 3^{512} **mod** $99 = 9$, and 3^{1024} **mod** $99 = 81$. Thus

our final answer will be the product of 3, 9, 36, 81, 27, 36, 9, and 81. We compute these one at a time modulo 99: $3 \cdot 9$ is 27, $27 \cdot 36$ is 81, $81 \cdot 81$ is 27, $27 \cdot 27$ is 36, $36 \cdot 36$ is 9, $9 \cdot 9$ is 81, and finally $81 \cdot 81$ is 27. So $3^{2003} \bmod 99 = 27$.

29. The binary expansion of an integer represents the integer as a sum of distinct powers of 2. For example, since $21 = (1\ 0101)_2$, we have $21 = 2^4 + 2^2 + 2^0$. Since binary expansions are unique, each integer can be so represented uniquely.

31. Let the decimal expansion of the integer a be given by $a = (a_{n-1}a_{n-2}\ldots a_1a_0)_{10}$. Thus $a = 10^{n-1}a_{n-1} + 10^{n-2}a_{n-2} + \cdots + 10a_1 + a_0$. Since $10 \equiv 1 \pmod 3$, we have $a \equiv a_{n-1} + a_{n-2} + \cdots + a_1 + a_0 \pmod 3$. Therefore $a \equiv 0 \pmod 3$ if and only if the sum of the digits is congruent to 0 (mod 3). Since being divisible by 3 is the same as being congruent to 0 (mod 3), we have proved that a positive integer is divisible by 3 if and only if the sum of its decimal digits is divisible by 3.

33. Let the binary expansion of the positive integer a be given by $a = (a_{n-1}a_{n-2}\ldots a_1a_0)_2$. Thus $a = a_0 + 2a_1 + 2^2a_2 + \cdots + 2^{n-1}a_{n-1}$. Since $2^2 \equiv 1 \pmod 3$, we see that $2^k \equiv 1 \pmod 3$ when k is even, and $2^k \equiv 2 \equiv -1 \pmod 3$ when k is odd. Therefore we have $a \equiv a_0 - a_1 + a_2 - a_3 + \cdots \pm a_{n-1} \pmod 3$. Thus $a \equiv 0 \pmod 3$ if and only if the sum of the binary digits in the even-numbered positions minus the sum of the binary digits in the odd-numbered positions is congruent to 0 modulo 3. Since being divisible by 3 is the same as being congruent to 0 (mod 3), our proof is complete.

35. **a)** Since the leading bit is a 1, this represents a negative number. The binary expansion of the absolute value of this number is the complement of the rest of the expansion, namely the complement of 1001, or 0110. Since $(0110)_2 = 6$, the answer is -6.

b) Since the leading bit is a 0, this represents a positive number, namely the number whose binary expansion is the rest of this string, 1101. Since $(1101)_2 = 13$, the answer is 13.

c) The answer is the negative of the complement of 0001, namely $-(1110)_2 = -14$.

d) $-(0000)_2 = 0$; note that 0 has two different representations, 0000 and 1111

37. We must assume that the sum actually represents a number in the appropriate range. Assume that n bits are being used, so that numbers strictly between -2^{n-1} and 2^{n-1} can be represented. The answer is almost, but not quite, that to obtain the one's complement representation of the sum of two numbers, we simply add the two strings representing these numbers using Algorithm 3. Instead, after performing this operation, there may be a carry out of the left-most column; in such a case, we then add 1 more to the answer. For example, suppose that $n = 4$; then numbers from -7 to 7 can be represented. To add -5 and 3, we add 1010 and 0011, obtaining 1101; there was no carry out of the left-most column. Since 1101 is the one's complement representation of -2, we have the correct answer. On the other hand, to add -4 and -3, we add 1011 and 1100, obtaining 1 0111. The 1 that was carried out of the left-most column is instead added to 0111, yielding 1000, which is the one's complement representation of -7. A proof that this method works entails considering the various cases determined by the signs and magnitudes of the addends.

39. If m is positive (or 0), then the leading bit (a_{n-1}) is 0, so the formula reads simply $m = \sum_{i=0}^{n-2} a_i 2^i$, which is clearly correct, since this is the binary expansion of m. (See Section 2.4 for the meaning of summation notation. This symbolism is a shorthand way of writing $a_0 + 2a_1 + 4a_2 + \cdots + 2^{n-2}a_{n-2}$.) Now suppose that m is negative. The one's complement expansion for m has its leading bit equal to 1. By the definition of one's complement, we can think of obtaining the remaining $n-1$ bits by subtracting $-m$, written in binary, from $111\ldots1$ (with $n-1$ 1's), since subtracting a bit from 1 is the same thing as complementing it. Equivalently, if we view the bit string $(a_{n-2}a_{n-1}\ldots a_0)$ as a binary number, then it represents $(2^{n-1} - 1) - (-m)$. In

symbols, this says that $(2^{n-1}-1)-(-m)=\sum_{i=0}^{n-2}a_i2^i$. Solving for m gives us the equation we are trying to prove (since $a_{n-1}=1$).

41. Following the definition, if the first bit is a 0, then we just evaluate the binary expansion. If the first bit is a 1, then we find what number x is represented by the remaining four bits in binary; the answer is then $-(2^4-x)$.

a) Since the first bit is a 1, and the remaining bits represent the number 9, this string represents the number $-(2^4-9)=-7$.

b) Since the first bit is a 0 and this is just the binary expansion of 13, the answer is 13.

c) Since the first bit is a 1, and the remaining bits represent the number 1, this string represents the number $-(2^4-1)=-15$.

d) Since the first bit is a 1, and the remaining bits represent the number 15, this string represents the number $-(2^4-15)=-1$. Note that 10000 would represent $-(2^4-0)=-16$, so in fact we can represent one extra negative number than positive number with this notation.

43. The nice thing about two's complement arithmetic is that we can just work as if it were all in base 2, since $-x$ (where x is positive) is represented by 2^n-x; in other words, modulo 2^n, negative numbers represent themselves. However, if overflow occurs, then we must recognize an error. Let us look at some examples, where $n=5$ (i.e., we use five bits to represent numbers between -15 and 15). To add $5+7$, we write $00101+00111=01100$ in base 2, which gives us the correct answer, 12. However, if we try to add $13+7$ we obtain $01101+00111=10100$, which represents -12, rather than 20, so we report an overflow error. (Of course these two numbers are congruent modulo 32.) Similarly, for $5+(-7)$, we write $00101+11001=11110$ in base 2, and 11110 is the two's complement representation of -2, the right answer. For $(-5)+(-7)$, we write $11011+11001=110100$ in base 2; if we ignore the extra 1 in the left-most column (which doesn't exist), then this is the two's complement representation of -12, again the right answer. To summarize, to obtain the two's complement representation of the sum of two integers given in two's complement representation, add them as if they were binary integers, and ignore any carry out of the left-most column. However, if the left-most digits of the two addends agree and the left-most digit of the answer is different from their common value, then an overflow has occurred, and the answer is not valid.

45. If m is positive (or 0), then the leading bit (a_{n-1}) is 0, so the formula reads simply $m=\sum_{i=0}^{n-2}a_i2^i$, which is clearly correct, since this is the binary expansion of m. (See Section 2.4 for the meaning of summation notation. This symbolism is a shorthand way of writing $a_0+2a_1+4a_2+\cdots+2^{n-2}a_{n-2}$.) Now suppose that m is negative. The two's complement expansion for m has its leading bit equal to 1. By the definition of two's complement, the remaining $n-1$ bits are the binary expansion of $2^{n-1}-(-m)$. In symbols, this says that $2^{n-1}-(-m)=\sum_{i=0}^{n-2}a_i2^i$. Solving for m gives us the equation we are trying to prove (since $a_{n-1}=1$).

47. Clearly we need $4n$ digits, four for each digit of the decimal representation.

49. To find the Cantor expansion, we will work from left to right. Thus the first step will be to find the largest number n whose factorial is still less than or equal to the given positive integer x. Then we determine the digits in the expansion, starting with a_n and ending with a_1.

procedure *Cantor*(x : positive integer)
$n := 1$; *factorial* $:= 1$
while $(n+1) \cdot$ *factorial* $\leq x$
    $n := n + 1$
    factorial $:=$ *factorial* $\cdot\, n$
{at this point we know that there are n digits in the expansion}
$y := x$ {this is just so we do not destroy the original input}
while $n > 0$
    $a_n := \lfloor y/\textit{factorial} \rfloor$
    $y := y - a_n \cdot$ *factorial*
    factorial $:=$ *factorial*$/n$
    $n := n - 1$
{we are done: $x = a_n n! + a_{n-1}(n-1)! + \cdots + a_2 2! + a_1 1!$}

51. Note that $n = 5$. Initially the carry is $c = 0$, and we start the **for** loop with $j = 0$. Since $a_0 = 1$ and $b_0 = 0$, we set d to be $\lfloor (1+0+0)/2 \rfloor = 0$; then $s_0 = 1+0+0-2\cdot 0$, which equals 1, and finally $c = 0$. At the end of the first pass, then, the right-most digit of the answer has been determined (it's a 1), and there is a carry of 0 into the next column.

Now $j = 1$, and we compute d to be $\lfloor (a_1 + b_1 + c)/2 \rfloor = \lfloor (1+1+0)/2 \rfloor = 1$; whereupon s_1 becomes $1+1+0-2\cdot 1 = 0$, and c is set to 1. Thus far we have determined that the last two bits of the answer are 01 (from left to right), and there is a carry of 1 into the next column.

The next three passes through the loop are similar. As a result of the pass when $j = 2$ we set $d = 1$, $s_2 = 0$, and then $c = 1$. When $j = 3$, we obtain $d = 1$, $s_3 = 0$, and then $c = 1$. Finally, when $j = 4$, we obtain $d = 1$, $s_4 = 1$, and then $c = 1$. At this point the loop is terminated, and when we execute the final step, $s_5 = 1$. Thus the answer is 11 0001.

53. We will assume that the answer is not negative, since otherwise we would need something like the one's complement representation. The algorithm is similar to the algorithm for addition, except that we need to borrow instead of carry. Rather than trying to incorporate the two cases (borrow or no borrow) into one, as was done in the algorithm for addition, we will use an **if... then** statement to treat the cases separately. The notation is the usual one: $a = (a_{n-1} \ldots a_1 a_0)_2$ and $b = (b_{n-1} \ldots b_1 b_0)_2$

procedure *subtract*(a, b : nonnegative integers)
borrow $:= 0$
for $j := 0$ **to** $n - 1$
    if $a_j -$ *borrow* $\geq b_j$ **then**
        $s_j := a_j -$ *borrow* $- b_j$
        borrow $:= 0$
    else
        $s_j := a_j + 2 -$ *borrow* $- b_j$
        borrow $:= 1$
{assuming $a \geq b$, we have $a - b = (s_{n-1} s_{n-2} \ldots s_1 s_0)_2$}

55. To determine which of two integers (we assume they are nonnegative), given in binary as $a = (a_{n-1} \ldots a_1 a_0)_2$ and $b = (b_{n-1} \ldots b_1 b_0)_2$, is larger, we need to compare digits from the most significant end ($i = n-1$) to the least ($i = 0$), stopping if and when we find a difference. For variety here we record the answer as a character string; in most applications it would probably be better to set *compare* to one of three code values (such as -1, 1, and 0) to indicate which of the three possibilities held.

```
procedure compare(a, b : nonnegative integers)
i := n − 1
while i > 0 and a_i = b_i
        i := i − 1
if a_i > b_i then answer := "a > b"
else if a_i < b_i then answer := "a < b"
else answer := "a = b"
return answer
```

57. There is one division for each pass through the **while** loop. Also, each pass generates one digit in the base b expansion. Thus the number of divisions equals the number of digits in the base b expansion of n. This is just $\lfloor \log_b n \rfloor + 1$ (for example, numbers from 10 to 99, inclusive, have common logarithms in the interval $[1, 2)$). Therefore exactly $\lfloor \log_b n \rfloor + 1$ divisions are required, and this is $O(\log n)$. (We are counting only the actual division operation in the statement $q := \lfloor q/b \rfloor$. If we also count the implied division in the statement $a_k := q \bmod b$, then there are twice as many as we computed here. The big-O estimate is the same, of course.)

59. The only time-consuming part of the algorithm is the **while** loop, which is iterated q times. The work done inside is a subtraction of integers no bigger than a, which has $\log a$ bits. The results now follows from Example 9.

SECTION 4.3 Primes and Greatest Common Divisors

The prime numbers are the building blocks for the natural numbers in terms of multiplication, just as the elements (like carbon, oxygen, or uranium) are the building blocks of all matter. Just as we can put two hydrogen atoms and one oxygen atom together to form water, every composite natural number is uniquely constructed by multiplying together prime numbers. Analyzing numbers in terms of their prime factorizations allows us to solve many problems, such as finding greatest common divisors. Prime numbers have fascinated people for millennia, and many easy-to-state questions about them remain unanswered. Students interested in pursuing these topics more should definitely consider taking a course in number theory.

1. In each case we can just use trial division up to the square root of the number being tested.

a) Since $21 = 3 \cdot 7$, we know that 21 is not prime.

b) Since $2 \not| 29$, $3 \not| 29$, and $5 \not| 29$, we know that 29 is prime. We needed to check for prime divisors only up to $\sqrt{29}$, which is less than 6.

c) Since $2 \not| 71$, $3 \not| 71$, $5 \not| 71$, and $7 \not| 71$, we know that 71 is prime.

d) Since $2 \not| 97$, $3 \not| 97$, $5 \not| 97$, and $7 \not| 97$, we know that 97 is prime.

e) Since $111 = 3 \cdot 37$, we know that 111 is not prime.

f) Since $143 = 11 \cdot 13$, we know that 143 is not prime.

3. In each case we can use trial division, starting with the smallest prime and increasing to the next prime once we find that a given prime no longer is a divisor of what is left. A calculator comes in handy. Alternatively, one could use a factor tree.

a) We note that 2 is a factor of 88, and the quotient upon division by 2 is 44. We divide by 2 again, and then again, leaving a quotient of 11. Since 11 is prime, we are done, and we have found the prime factorization: $88 = 2^3 \cdot 11$.

b) $126 = 2 \cdot 63 = 2 \cdot 3 \cdot 21 = 2 \cdot 3 \cdot 3 \cdot 7 = 2 \cdot 3^2 \cdot 7$

c) $729 = 3 \cdot 243 = 3 \cdot 3 \cdot 81 = 3 \cdot 3 \cdot 3 \cdot 27 = 3 \cdot 3 \cdot 3 \cdot 3 \cdot 9 = 3 \cdot 3 \cdot 3 \cdot 3 \cdot 3 \cdot 3 = 3^6$

d) $1001 = 7 \cdot 143 = 7 \cdot 11 \cdot 13$

e) $1111 = 11 \cdot 101$ (we know that 101 is prime because we have already tried all prime factors less than $\sqrt{101}$)
f) $909090 = 2 \cdot 454545 = 2 \cdot 3 \cdot 151515 = 2 \cdot 3 \cdot 3 \cdot 50505 = 2 \cdot 3 \cdot 3 \cdot 3 \cdot 16835 = 2 \cdot 3 \cdot 3 \cdot 3 \cdot 5 \cdot 3367 = 2 \cdot 3 \cdot 3 \cdot 3 \cdot 5 \cdot 7 \cdot 481 = 2 \cdot 3 \cdot 3 \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot 37 = 2 \cdot 3^3 \cdot 5 \cdot 7 \cdot 13 \cdot 37$

5. $10! = 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 \cdot 9 \cdot 10 = 2 \cdot 3 \cdot 2^2 \cdot 5 \cdot (2 \cdot 3) \cdot 7 \cdot 2^3 \cdot 3^2 \cdot (2 \cdot 5) = 2^8 \cdot 3^4 \cdot 5^2 \cdot 7$

7. The input is an integer n greater than 1. We try dividing it by all integers from 2 to $\sqrt{n}$, and if we find one that leaves no remainder then we know that n is not prime. The pseudocode below accomplishes this.

```
procedure primetester(n : integer greater than 1)
isprime := true
d := 2
while isprime and d ≤ √n
        if n mod d = 0 then isprime := false
        else d := d + 1
return isprime
```

9. We use what we know about factoring from algebra. In particular, we know that $a^m + 1 = (a+1)(a^{m-1} - a^{m-2} + a^{m-3} + \cdots - 1)$. (Notice that this works if and only if m is odd, because the final sign has to be a plus sign.) Because a and m are both greater than 1, we know that $1 < a + 1 < a^m + 1$. This provides a factoring of $a^m + 1$ into two proper factors, so $a^m + 1$ is composite.

11. We give a proof by contradiction. Suppose that in fact $\log_2 3$ is the rational number p/q, where p and q are integers. Since $\log_2 3 > 0$, we can assume that p and q are positive. Translating the equation $\log_2 3 = p/q$ into its exponential equivalent, we obtain $3 = 2^{p/q}$. Raising both sides to the q^{th} power yields $3^q = 2^p$. Now this is a violation of the Fundamental Theorem of Arithmetic, since it gives two different prime factorizations of the same number. Hence our assumption (that $\log_2 3$ is rational) must be wrong, and we conclude that $\log_2 3$ is irrational.

13. This is simply an existence statement. To prove that it is true, we need only exhibit the primes. Indeed, 3, 5, and 7 satisfy the conditions. (Actually, this is the only example, and a harder problem is to prove that there are no others.)

15. The prime factors of 30 are 2, 3, and 5. Thus we are looking for positive integers less than 30 that have none of these as prime factors. Since the smallest prime number other than these is 7, and 7^2 is already greater than 30, in fact only primes (and the number 1) will satisfy this condition. Therefore the answer is 1, 7, 11, 13, 17, 19, 23, and 29.

17. **a)** Since $\gcd(11, 15) = 1$, $\gcd(11, 19) = 1$, and $\gcd(15, 19) = 1$, these three numbers are pairwise relatively prime.
b) Since $\gcd(15, 21) = 3 > 1$, these three numbers are not pairwise relatively prime.
c) Since $\gcd(12, 17) = 1$, $\gcd(12, 31) = 1$, $\gcd(12, 37) = 1$, $\gcd(17, 31) = 1$, $\gcd(17, 37) = 1$, and $\gcd(31, 37) = 1$, these four numbers are pairwise relatively prime. (Indeed, the last three are primes, and the prime factors of the first are 2 and 3.)
d) Again, since no two of 7, 8, 9, and 11 have a common factor greater than 1, this set is pairwise relatively prime.

19. The identity shown in the hint is valid, as can be readily seen by multiplying out the right-hand side (all the terms cancel—telescope—except for 2^{ab} and -1). We will prove the assertion by proving its contrapositive. Suppose that n is *not* prime. Then by definition $n = ab$ for some integers a and b each greater than 1. Since $a > 1$, $2^a - 1$, the first factor in the suggested identity, is greater than 1. Clearly the second factor is greater than 1. Thus $2^n - 1 = 2^{ab} - 1$ is the product of two integers each greater than 1, so it is not prime.

21. We compute $\phi(n)$ here by enumerating the set of positive integers less than n that are relatively prime to n.
a) $\phi(4) = |\{1,3\}| = 2$ **b)** $\phi(10) = |\{1,3,7,9\}| = 4$
c) $\phi(13) = |\{1,2,3,4,5,6,7,8,9,10,11,12\}| = 12$

23. All the positive integers less than or equal to p^k (and there are clearly p^k of them) are less than p^k and relatively prime to p^k unless they are a multiple of p. Since the fraction $1/p$ of them are multiples of p, we have $\phi(p^k) = p^k(1 - 1/p) = p^k - p^{k-1}$.

25. To find the greatest common divisor of two numbers whose prime factorizations are given, we just need to take the smaller exponent for each prime.
a) The first number has no prime factors of 2, so the gcd has no 2's. Since the first number has seven factors of 3, but the second number has only five, the gcd has five factors of 3. Similarly the gcd has a factor of 5^3. So the gcd is $3^5 \cdot 5^3$.
b) These numbers have no common prime factors, so the gcd is 1. **c)** 23^{17} **d)** $41 \cdot 43 \cdot 53$
e) These numbers have no common prime factors, so the gcd is 1.
f) The gcd of any positive integer and 0 is that integer, so the answer is 1111.

27. To find the least common multiple of two numbers whose prime factorizations are given, we just need to take the larger exponent for each prime.
a) The first number has no prime factors of 2 but the second number has 11 of them, so the lcm has 11 factors of 2. Since the first number has seven factors of 3 and the second number has five, the lcm has seven factors of 3. Similarly the lcm has a factor of 5^9 and a factor of 7^3. So the lcm is $2^{11} \cdot 3^7 \cdot 5^9 \cdot 7^3$.
b) These numbers have no common prime factors, so the lcm is their product, $2^9 \cdot 3^7 \cdot 5^5 \cdot 7^3 \cdot 11 \cdot 13 \cdot 17$.
c) 23^{31} **d)** $41 \cdot 43 \cdot 53$ **e)** $2^{12} \cdot 3^{13} \cdot 5^{17} \cdot 7^{21}$, as in part **(b)**
f) It makes no sense to ask for a positive multiple of 0, so this question has no answer. Least common multiples are defined only for positive integers.

29. First we find the prime factorizations: $92928 = 2^8 \cdot 3 \cdot 11^2$ and $123552 = 2^5 \cdot 3^3 \cdot 11 \cdot 13$. Therefore $\gcd(92928, 123552) = 2^5 \cdot 3 \cdot 11 = 1056$ and $\text{lcm}(92928, 123552) = 2^8 \cdot 3^3 \cdot 11^2 \cdot 13 = 10872576$. The requested products are $(2^5 \cdot 3 \cdot 11) \cdot (2^8 \cdot 3^3 \cdot 11^2 \cdot 13)$ and $(2^8 \cdot 3 \cdot 11^2) \cdot (2^5 \cdot 3^3 \cdot 11 \cdot 13)$, both of which are $2^{13} \cdot 3^4 \cdot 11^3 \cdot 13 = 11{,}481{,}440{,}256$.

31. The important observation to make here is that the smaller of any two numbers plus the larger of the two numbers is always equal to the sum of the two numbers. Since the exponent of the prime p in $\gcd(a,b)$ is the smaller of the exponents of p in a and in b, and since the exponent of the prime p in $\text{lcm}(a,b)$ is the larger of the exponents of p in a and in b, the exponent of p in $\gcd(a,b)\text{lcm}(a,b)$ is the sum of the smaller and the larger of these two values. Therefore by the observation, it equals the sum of the two values themselves, which is clearly equal to the exponent of p in ab. Since this is true for every prime p, we conclude that $\gcd(a,b)\text{lcm}(a,b)$ and ab have the same prime factorizations and are therefore equal.

33. **a)** By Lemma 1, $\gcd(12,18)$ is the same as the gcd of the smaller of these two numbers (12) and the remainder when the larger (18) is divided by the smaller. In this case the remainder is 6, so $\gcd(12,18) = \gcd(12,6)$. Now $\gcd(12,6)$ is the same as the gcd of the smaller of *these* two numbers (6) and the remainder when the larger (12) is divided by the smaller, namely 0. This gives $\gcd(12,6) = \gcd(6,0)$. But $\gcd(x,0) = x$ for all positive integers, so $\gcd(6,0) = 6$. Thus the answer is 6. In brief (the form we will use for the remaining parts), $\gcd(12,18) = \gcd(12,6) = \gcd(6,0) = 6$.
b) $\gcd(111,201) = \gcd(111,90) = \gcd(90,21) = \gcd(21,6) = \gcd(6,3) = \gcd(3,0) = 3$

c) $\gcd(1001, 1331) = \gcd(1001, 330) = \gcd(330, 11) = \gcd(11, 0) = 11$

d) $\gcd(12345, 54321) = \gcd(12345, 4941) = \gcd(4941, 2463) = \gcd(2463, 15) = \gcd(15, 3) = \gcd(3, 0) = 3$

e) $\gcd(1000, 5040) = \gcd(1000, 40) = \gcd(40, 0) = 40$

f) $\gcd(9888, 6060) = \gcd(6060, 3828) = \gcd(3828, 2232) = \gcd(2232, 1596) = \gcd(1596, 636) = \gcd(636, 324) = \gcd(324, 312) = \gcd(312, 12) = \gcd(12, 0) = 12$

35. In carrying out the Euclidean algorithm on this data, we divide successively by 55, 34, 21, 13, 8, 5, 3, 2, and 1, so nine divisions are required.

37. One can compute $\gcd(2^a - 1, 2^b - 1)$ using the Euclidean algorithm. Let us look at what happens when we do so. If $b = 1$, then the answer is just 1, which is the same as $2^{\gcd(a,b)} - 1$ in this case. Otherwise, we reduce the problem to computing $\gcd(2^b - 1, (2^a - 1) \bmod (2^b - 1))$. Now from Exercise 36 we know that this second argument equals $2^{a \bmod b} - 1$. Therefore the exponents involved in the continuing calculation are b and $a \bmod b$—exactly the same quantities that are involved in computing $\gcd(a, b)$! It follows that when the process terminates, the answer must be $2^{\gcd(a,b)} - 1$, as desired.

39. **a)** This first one is easy to do by inspection. Clearly 10 and 11 are relatively prime, so their greatest common divisor is 1, and $1 = 11 - 10 = (-1) \cdot 10 + 1 \cdot 11$.

b) In order to find the coefficients s and t such that $21s + 44t = \gcd(21, 44)$, we carry out the steps of the Euclidean algorithm.

$$44 = 2 \cdot 21 + 2$$
$$21 = 10 \cdot 2 + 1$$

Then we work up from the bottom, expressing the greatest common divisor (which we have just seen to be 1) in terms of the numbers involved in the algorithm, namely 44, 21, and 2. In particular, the last equation tells us that $1 = 21 - 10 \cdot 2$, so that we have expressed the gcd as a linear combination of 21 and 2. But now the first equation tells us that $2 = 44 - 2 \cdot 21$; we plug this into our previous equation and obtain

$$1 = 21 - 10 \cdot (44 - 2 \cdot 21) = 21 \cdot 21 - 10 \cdot 44.$$

Thus we have expressed 1 as a linear combination (with integer coefficients) of 21 and 44, namely $\gcd(21, 44) = 21 \cdot 21 + (-10) \cdot 44$.

c) Again, we carry out the Euclidean algorithm. Since $48 = 1 \cdot 36 + 12$, and $12 \mid 36$, we know that $\gcd(36, 48) = 12$. From the equation shown here, we can immediately write $12 = (-1) \cdot 36 + 48$.

d) The calculation of the greatest common divisor takes several steps:

$$55 = 1 \cdot 34 + 21$$
$$34 = 1 \cdot 21 + 13$$
$$21 = 1 \cdot 13 + 8$$
$$13 = 1 \cdot 8 + 5$$
$$8 = 1 \cdot 5 + 3$$
$$5 = 1 \cdot 3 + 2$$
$$3 = 1 \cdot 2 + 1$$

Then we need to work our way back up, successively plugging in for the remainders determined in this

calculation:

$$\begin{aligned}1 &= 3 - 2\\ &= 3 - (5 - 3) = 2 \cdot 3 - 5\\ &= 2 \cdot (8 - 5) - 5 = 2 \cdot 8 - 3 \cdot 5\\ &= 2 \cdot 8 - 3 \cdot (13 - 8) = 5 \cdot 8 - 3 \cdot 13\\ &= 5 \cdot (21 - 13) - 3 \cdot 13 = 5 \cdot 21 - 8 \cdot 13\\ &= 5 \cdot 21 - 8 \cdot (34 - 21) = 13 \cdot 21 - 8 \cdot 34\\ &= 13 \cdot (55 - 34) - 8 \cdot 34 = 13 \cdot 55 - 21 \cdot 34\end{aligned}$$

e) Here are the two calculations—down to the gcd using the Euclidean algorithm, and then back up by substitution until we have expressed the gcd as the desired linear combination of the original numbers.

$$\begin{aligned}213 &= 1 \cdot 117 + 96\\ 117 &= 1 \cdot 96 + 21\\ 96 &= 4 \cdot 21 + 12\\ 21 &= 1 \cdot 12 + 9\\ 12 &= 1 \cdot 9 + 3\end{aligned}$$

Since $3 \mid 9$, we have $\gcd(117, 213) = 3$.

$$\begin{aligned}3 &= 12 - 9\\ &= 12 - (21 - 12) = 2 \cdot 12 - 21\\ &= 2 \cdot (96 - 4 \cdot 21) - 21 = 2 \cdot 96 - 9 \cdot 21\\ &= 2 \cdot 96 - 9 \cdot (117 - 96) = 11 \cdot 96 - 9 \cdot 117\\ &= 11 \cdot (213 - 117) - 9 \cdot 117 = 11 \cdot 213 - 20 \cdot 117\end{aligned}$$

f) Clearly $\gcd(0, 223) = 223$, so we can write $223 = s \cdot 0 + 1 \cdot 223$ for any integer s.

g) Here are the two calculations—down to the gcd using the Euclidean algorithm, and then back up by substitution until we have expressed the gcd as the desired linear combination of the original numbers.

$$\begin{aligned}2347 &= 19 \cdot 123 + 10\\ 123 &= 12 \cdot 10 + 3\\ 10 &= 3 \cdot 3 + 1\end{aligned}$$

Thus the greatest common divisor is 1.

$$\begin{aligned}1 &= 10 - 3 \cdot 3\\ &= 10 - 3 \cdot (123 - 12 \cdot 10) = 37 \cdot 10 - 3 \cdot 123\\ &= 37 \cdot (2347 - 19 \cdot 123) - 3 \cdot 123 = 37 \cdot 2347 - 706 \cdot 123\end{aligned}$$

h) Here are the two calculations—down to the gcd using the Euclidean algorithm, and then back up by substitution until we have expressed the gcd as the desired linear combination of the original numbers.

$$\begin{aligned}4666 &= 3454 + 1212\\ 3454 &= 2 \cdot 1212 + 1030\\ 1212 &= 1030 + 182\\ 1030 &= 5 \cdot 182 + 120\\ 182 &= 120 + 62\\ 120 &= 62 + 58\\ 62 &= 58 + 4\\ 58 &= 14 \cdot 4 + 2\end{aligned}$$

Since $2 \mid 4$, the greatest common divisor is 2.

$$\begin{aligned}
2 &= 58 - 14 \cdot 4 \\
&= 58 - 14 \cdot (62 - 58) = 15 \cdot 58 - 14 \cdot 62 \\
&= 15 \cdot (120 - 62) - 14 \cdot 62 = 15 \cdot 120 - 29 \cdot 62 \\
&= 15 \cdot 120 - 29 \cdot (182 - 120) = 44 \cdot 120 - 29 \cdot 182 \\
&= 44 \cdot (1030 - 5 \cdot 182) - 29 \cdot 182 = 44 \cdot 1030 - 249 \cdot 182 \\
&= 44 \cdot 1030 - 249 \cdot (1212 - 1030) = 293 \cdot 1030 - 249 \cdot 1212 \\
&= 293 \cdot (3454 - 2 \cdot 1212) - 249 \cdot 1212 = 293 \cdot 3454 - 835 \cdot 1212 \\
&= 293 \cdot 3454 - 835 \cdot (4666 - 3454) = 1128 \cdot 3454 - 835 \cdot 4666
\end{aligned}$$

i) Here are the two calculations—down to the gcd using the Euclidean algorithm, and then back up by substitution until we have expressed the gcd as the desired linear combination of the original numbers.

$$\begin{aligned}
11111 &= 9999 + 1112 \\
9999 &= 8 \cdot 1112 + 1103 \\
1112 &= 1103 + 9 \\
1103 &= 122 \cdot 9 + 5 \\
9 &= 5 + 4 \\
5 &= 4 + 1
\end{aligned}$$

Thus 1 is the greatest common divisor.

$$\begin{aligned}
1 &= 5 - 4 \\
&= 5 - (9 - 5) = 2 \cdot 5 - 9 \\
&= 2 \cdot (1103 - 122 \cdot 9) - 9 = 2 \cdot 1103 - 245 \cdot 9 \\
&= 2 \cdot 1103 - 245 \cdot (1112 - 1103) = 247 \cdot 1103 - 245 \cdot 1112 \\
&= 247 \cdot (9999 - 8 \cdot 1112) - 245 \cdot 1112 = 247 \cdot 9999 - 2221 \cdot 1112 \\
&= 247 \cdot 9999 - 2221 \cdot (11111 - 9999) = 2468 \cdot 9999 - 2221 \cdot 11111
\end{aligned}$$

41. When we apply the Euclidean algorithm we obtain the following quotients and remainders: $q_1 = 0$, $r_2 = 26$, $q_2 = 3$, $r_3 = 13$, $q_3 = 2$. Note that $n = 3$. Thus we compute the successive s's and t's as follows, using the given recurrences:

$s_2 = s_0 - q_1 s_1 = 1 - 0 \cdot 0 = 1,$ $\qquad$ $t_2 = t_0 - q_1 t_1 = 0 - 0 \cdot 1 = 0$

$s_3 = s_1 - q_2 s_2 = 0 - 3 \cdot 1 = -3,$ $\qquad$ $t_3 = t_1 - q_2 t_2 = 1 - 3 \cdot 0 = 1$

Thus we have $s_3 a + t_3 b = (-3) \cdot 26 + 1 \cdot 91 = 13$, which is $\gcd(26, 91)$.

43. When we apply the Euclidean algorithm we obtain the following quotients and remainders: $q_1 = 1$, $r_2 = 55$, $q_2 = 1$, $r_3 = 34$, $q_3 = 1$, $r_4 = 21$, $q_4 = 1$, $r_5 = 13$, $q_5 = 1$, $r_6 = 8$, $q_6 = 1$, $r_7 = 5$, $q_7 = 1$, $r_8 = 3$, $q_8 = 1$, $r_9 = 2$, $q_9 = 1$, $r_{10} = 1$, $q_{10} = 2$. Note that $n = 10$. Thus we compute the successive s's and t's as follows, using the given recurrences:

$s_2 = s_0 - q_1 s_1 = 1 - 1 \cdot 0 = 1,$ $\qquad$ $t_2 = t_0 - q_1 t_1 = 0 - 1 \cdot 1 = -1$

$s_3 = s_1 - q_2 s_2 = 0 - 1 \cdot 1 = -1,$ $\qquad$ $t_3 = t_1 - q_2 t_2 = 1 - 1 \cdot (-1) = 2$

$s_4 = s_2 - q_3 s_3 = 1 - 1 \cdot (-1) = 2,$ $\qquad$ $t_4 = t_2 - q_3 t_3 = -1 - 1 \cdot 2 = -3$

$s_5 = s_3 - q_4 s_4 = -1 - 1 \cdot 2 = -3,$ $\qquad$ $t_5 = t_3 - q_4 t_4 = 2 - 1 \cdot (-3) = 5$

$s_6 = s_4 - q_5 s_5 = 2 - 1 \cdot (-3) = 5,$ $\qquad$ $t_6 = t_4 - q_5 t_5 = -3 - 1 \cdot 5 = -8$

$s_7 = s_5 - q_6 s_6 = -3 - 1 \cdot 5 = -8,$ $\qquad$ $t_7 = t_5 - q_6 t_6 = 5 - 1 \cdot (-8) = 13$

$s_8 = s_6 - q_7 s_7 = 5 - 1 \cdot (-8) = 13$, $t_8 = t_6 - q_7 t_7 = -8 - 1 \cdot 13 = -21$
$s_9 = s_7 - q_8 s_8 = -8 - 1 \cdot 13 = -21$, $t_9 = t_7 - q_8 t_8 = 13 - 1 \cdot (-21) = 34$
$s_{10} = s_8 - q_9 s_9 = 13 - 1 \cdot (-21) = 34$, $t_{10} = t_8 - q_9 t_9 = -21 - 1 \cdot 34 = -55$

Thus we have $s_{10}a + t_{10}b = 34 \cdot 144 + (-55) \cdot 89 = 1$, which is $\gcd(144, 89)$.

45. We start with the pseudocode for the Euclidean algorithm (Algorithm 1) and add variables to keep track of the s and t values. We need three of them, since the new s depends on the previous two s's, and similarly for t. We also need to keep track of q.

procedure *extended Euclidean*(a, b : positive integers)
$x := a$
$y := b$
$oldolds := 1$
$olds := 0$
$oldoldt := 0$
$oldt := 1$
while $y \neq 0$
    $q := x$ **div** y
    $r := x$ **mod** y
    $x := y$
    $y := r$
    $s := oldolds - q \cdot olds$
    $t := oldoldt - q \cdot oldt$
    $oldolds := olds$
    $oldoldt := oldt$
    $olds := s$
    $oldt := t$
{ $\gcd(a, b)$ is x, and the Bézout coefficients are given by $(oldolds)a + (oldoldt)b = x$ }

47. Obviously there are no definitive answers to these problems, but we present below a reasonable and satisfying rule for forming the sequence in each case.

a) There are 1's in the prime locations and 0's elsewhere. In other words, the n^{th} term of the sequence is 1 if n is a prime number and 0 otherwise.

b) The suspicious 2's occurring every other term and the appearance of the 11 and 13 lead us to discover that the n^{th} term is the smallest prime factor of n (and is 1 when $n = 1$).

c) The n^{th} term is the number of positive divisors of n. For example, the twelfth term is 6, since 12 has the positive divisors 1, 2, 3, 4, 6, and 12. A tip-off to get us going in the right direction is that there are 2's in the prime locations.

d) Perhaps the composer of the problem had something else in mind, but one rule here is that the n^{th} term is 0 if and only if n has a repeated prime factor; the 1's occur at locations for which n is "square-free" (has no factor, other than 1, that is a perfect square). For example, 12 has the square 2^2, so the twelfth term is 0.

e) We note that all the terms (after the first one) are primes. This leads us to guess that the n^{th} term is the largest prime less than or equal to n (and is 1 when $n = 1$).

f) Each term comes from the one before it by multiplying by a certain number. The multipliers are 2, 3, 5, 7, 11, 13, 17, 19, and 23—the primes. So the rule seems to be that we obtain the next term from the n^{th} term by multiplying by the n^{th} prime number (and we start at 1). In other words, the n^{th} term is the product of the smallest $n - 1$ prime numbers.

49. Consider the product $n(n+1)(n+2)$ for some integer n. Since every second integer is even (divisible by 2), this product is divisible by 2. Since every third integer is divisible by 3, this product is divisible by 3. Therefore this product has both 2 and 3 in its prime factorization and is therefore divisible by $2 \cdot 3 = 6$.

51. It is hard to know how to get started on this problem. To some extent, mathematics is an experimental science, so it would probably be a good idea to compute $n^2 - 79n + 1601$ for several positive integer values of n to get a feel for what is happening. Using a computer, or at least a calculator, would be helpful. If we plug in $n = 1, 2, 3, 4,$ and 5, then we get the values 1523, 1447, 1373, 1301, and 1231, all of which are prime. This may lead us to believe that the proposition is true, but it gives us no clue as to how to prove it. Indeed, it seems as if it would be very hard to prove that this expression *always* produces a prime number, since being prime means the absence of nontrivial factors, and nothing in the expression seems to be very helpful in proving such a negative assertion. (The fact that we cannot factor it algebraically is irrelevant—in fact, if it factored algebraically, then it would essentially *never* be prime.) Perhaps we should try some more integers. If we do so, we find a lot more prime numbers, but we are still skeptical. Well, perhaps there is some way to arrange that this expression will have a factor. How about 1601? Well, yes! If we let $n = 1601$, then all three terms will have 1601 as a common factor, so that 1601 is a factor of the entire expression. In fact, $1601^2 - 79 \cdot 1601 + 1601 = 1601 \cdot 1523$. So we have found a counterexample after all, and the proposition is false. Note that this was not a problem in which we could proceed in a calm, calculated way from problem to solution. Mathematics is often like that—lots of false leads and approaches that get us nowhere, and then suddenly a burst of insight that solves the problem. (The smallest n for which this expression is not prime is $n = 80$; this gives the value $1681 = 41 \cdot 41$.)

53. Here is one way to find a composite term in the sequence. If we set $k = 1$, then we get $a + b$. That number is greater than 1, but it may not be composite. So let's increase k by $a + b$, which will have the effect of adding a multiple of $a + b$ to our previous answer, and we will therefore get a composite number, because $a + b$ will be a nontrivial factor of it. So setting $k = a + b + 1$ should work. Indeed, with that choice we have $ak + b = a(a + b + 1) + b = a^2 + ab + a + b$, which factors nicely as $(a + 1)(a + b)$. Since a and b are both positive integers, both factors are greater than 1, and we have our composite number.

55. Recall that the proof that there are infinitely many primes starts by assuming that there are only finitely many primes $p_1, p_2, \ldots, p_n$, and forming the number $p_1 p_2 \cdots p_n + 1$. This number is either prime or has a prime factor different from each of the primes $p_1, p_2, \ldots, p_n$; this shows that there are infinitely many primes. So, let us suppose that there are only finitely many primes of the form $4k+3$, namely $q_1, q_2, \ldots, q_n$, where $q_1 = 3$, $q_2 = 7$, and so on.

What number can we form that is not divisible by any of these primes, but that must be divisible by a prime of the form $4k + 3$? We might consider the number $4q_1 q_2 \cdots q_n + 3$. Unfortunately, this number is not prime, as it is is divisible by 3 (because $q_1 = 3$). Instead we consider the number $Q = 4q_1 q_2 \cdots q_n - 1$. Note that Q is of the form $4k + 3$ (where $k = q_1 q_2 \cdots q_n - 1$). If Q is prime, then we have found a prime of the desired form different from all those listed. If Q is not prime, then Q has at least one prime factor not in the list $q_1, q_2, \ldots, q_n$, because the remainder when Q is divided by q_j is $q_j - 1$, and $q_j - 1 \neq 0$. Therefore $q_j \nmid Q$ for $j = 1, 2, \ldots, n$. Because all odd primes are either of the form $4k + 1$ or of the form $4k + 3$, and the product of primes of the form $4k + 1$ is also of this form (because $(4k + 1)(4m + 1) = 4(4km + k + m) + 1$), there must be a factor of Q of the form $4k + 3$ different from the primes we listed. This complete the proof.

57. We need to show that this function is one-to-one and onto. In other words, if we are given a positive integer x, we must show that there is exactly one positive rational number m/n (written in lowest terms) such that $K(m/n) = x$. To do this, we factor x into its prime factorization and then read off the m and n such that $K(m/n) = x$. The primes that occur to even powers are the primes that occur in the prime factorization of m, with the exponents being half the corresponding exponents in x; and the primes that occur to odd powers are the primes that occur in the prime factorization of n, with the exponents being half of one more than the exponents in x. Since this uniquely determines m and n, there is one and only one fraction, in

lowest terms, that maps to x under K.

SECTION 4.4 Solving Congruences

Many of these exercises are reasonably straightforward calculations, but the amount of arithmetic involved in some of them can be formidable. Look at the worked out examples in the text if you need help getting the hang of it. The theoretical exercises, such as #18 and #19 give you a good taste of the kinds of proofs in an elementary number theory course.

1. We simply need to show that $15 \cdot 7 \equiv 1 \pmod{26}$, or in other words, that $15 \cdot 7 - 1$ is divisible by 26. But this quantity is 104, which is $26 \cdot 4$.

3. We want to find an integer k such that $4k$ is 1 greater than a multiple of 9. We compute $4 \cdot 1 = 4 = 0 \cdot 9 + 4$, $4 \cdot 2 = 8 = 0 \cdot 9 + 8$, $4 \cdot 3 = 12 = 1 \cdot 9 + 3$, $4 \cdot 4 = 16 = 1 \cdot 9 + 7$, $4 \cdot 5 = 20 = 2 \cdot 9 + 2$, $4 \cdot 6 = 24 = 2 \cdot 9 + 6$, $4 \cdot 7 = 28 = 3 \cdot 9 + 1$. Therefore an inverse of 4 modulo 9 is 7.

5. **a)** Following the procedure of Example 2, we carry out the Euclidean algorithm to find $\gcd(4, 9)$:

$$9 = 2 \cdot 4 + 1$$
$$4 = 4 \cdot 1$$

Then we work backwards to rewrite the gcd (the last nonzero remainder, which is 1 here) in terms of 4 and 9:

$$1 = 9 - 2 \cdot 4$$

Therefore the Bézout coefficients of 9 and 4 are 1 and -2, respectively. The coefficient of 4 is our desired answer, namely -2, which is the same as 7 modulo 9. Note that this agrees with our answer in Exercise 3.

b) We proceed as above:

$$141 = 7 \cdot 19 + 8$$
$$19 = 2 \cdot 8 + 3$$
$$8 = 2 \cdot 3 + 2$$
$$3 = 1 \cdot 2 + 1$$
$$2 = 2 \cdot 1$$

Then we work backwards to rewrite the gcd (the last nonzero remainder, which is 1 here) in terms of 141 and 19:

$$\begin{aligned} 1 &= 3 - 1 \cdot 2 \\ &= 3 - 1 \cdot (8 - 2 \cdot 3) = 3 \cdot 3 - 1 \cdot 8 \\ &= 3 \cdot (19 - 2 \cdot 8) - 1 \cdot 8 = 3 \cdot 19 - 7 \cdot 8 \\ &= 3 \cdot 19 - 7 \cdot (141 - 7 \cdot 19) = (-7) \cdot 141 + 52 \cdot 19 \end{aligned}$$

Therefore the Bézout coefficient of 19 is 52, and that is an inverse of 19 modulo 141.

c) We proceed as above:

$$89 = 1 \cdot 55 + 34$$
$$55 = 1 \cdot 34 + 21$$
$$34 = 1 \cdot 21 + 13$$
$$21 = 1 \cdot 13 + 8$$
$$13 = 1 \cdot 8 + 5$$
$$8 = 1 \cdot 5 + 3$$
$$5 = 1 \cdot 3 + 2$$
$$3 = 1 \cdot 2 + 1$$
$$2 = 1 \cdot 2$$

Then we work backwards to rewrite the gcd (the last nonzero remainder, which is 1 here) in terms of 89 and 55:

$$\begin{aligned}
1 &= 3 - 1 \cdot 2 \\
&= 3 - 1 \cdot (5 - 1 \cdot 3) = 2 \cdot 3 - 1 \cdot 5 \\
&= 2 \cdot (8 - 1 \cdot 5) - 1 \cdot 5 = 2 \cdot 8 - 3 \cdot 5 \\
&= 2 \cdot 8 - 3 \cdot (13 - 1 \cdot 8) = 5 \cdot 8 - 3 \cdot 13 \\
&= 5 \cdot (21 - 1 \cdot 13) - 3 \cdot 13 = 5 \cdot 21 - 8 \cdot 13 \\
&= 5 \cdot 21 - 8 \cdot (34 - 1 \cdot 21) = 13 \cdot 21 - 8 \cdot 34 \\
&= 13 \cdot (55 - 1 \cdot 34) - 8 \cdot 34 = 13 \cdot 55 - 21 \cdot 34 \\
&= 13 \cdot 55 - 21 \cdot (89 - 1 \cdot 55) = 34 \cdot 55 - 21 \cdot 89
\end{aligned}$$

Therefore the Bézout coefficient of 55 is 34, and that is an inverse of 55 modulo 89.

d) We proceed as above:

$$\begin{aligned}
232 &= 2 \cdot 89 + 54 \\
89 &= 1 \cdot 54 + 35 \\
54 &= 1 \cdot 35 + 19 \\
35 &= 1 \cdot 19 + 16 \\
19 &= 1 \cdot 16 + 3 \\
16 &= 5 \cdot 3 + 1 \\
3 &= 3 \cdot 1
\end{aligned}$$

Then we work backwards to rewrite the gcd (the last nonzero remainder, which is 1 here) in terms of 232 and 89:

$$\begin{aligned}
1 &= 16 - 5 \cdot 3 \\
&= 16 - 5 \cdot (19 - 1 \cdot 16) = 6 \cdot 16 - 5 \cdot 19 \\
&= 6 \cdot (35 - 1 \cdot 19) - 5 \cdot 19 = 6 \cdot 35 - 11 \cdot 19 \\
&= 6 \cdot 35 - 11 \cdot (54 - 1 \cdot 35) = 17 \cdot 35 - 11 \cdot 54 \\
&= 17 \cdot (89 - 1 \cdot 54) - 11 \cdot 54 = 17 \cdot 89 - 28 \cdot 54 \\
&= 17 \cdot 89 - 28 \cdot (232 - 2 \cdot 89) = 73 \cdot 89 - 28 \cdot 232
\end{aligned}$$

Therefore the Bézout coefficient of 89 is 73, and that is an inverse of 89 modulo 232.

7. We follow the hint. Suppose that we had two inverses of a modulo m, say b and c. In symbols, we would have $ba \equiv 1 \pmod{m}$ and $ca \equiv 1 \pmod{m}$. The first congruence says that m divides $ba - 1$, and the second says that m divides $ca - 1$. Therefore m divides the difference $(ba - 1) - (ca - 1) = ba - ca$. (The difference of two multiples of m is a multiple of m.) Thus $ba \equiv ca \pmod{m}$. It follows immediately from Theorem 7 in Section 4.3 (the roles of a, b, and c need to be permuted) that $b \equiv c \pmod{m}$, which is what we wanted to prove.

9. In Exercise 5a we found that an inverse of 4 modulo 9 is 7. Therefore we multiply both sides of this equation by 7, obtaining $x \equiv 35 \equiv 8 \pmod{9}$. As a check, we compute $4 \cdot 8 = 32 \equiv 5 \pmod{9}$.

11. Our answers are not unique, of course—anything in the same congruence class works just as well.

a) In Exercise 5b we found that an inverse of 19 modulo 141 is 52. Therefore we multiply both sides of this equation by 52, obtaining $x \equiv 208 \equiv 67 \pmod{141}$. As a check, we compute $19 \cdot 67 = 1273 \equiv 4 \pmod{141}$.

b) In Exercise 5c we found that an inverse of 55 modulo 89 is 34. Therefore we multiply both sides of this equation by 34, obtaining $x \equiv 1156 \equiv 88 \pmod{89}$. As a check, we compute $55 \cdot 88 \equiv 55 \cdot (-1) = -55 \equiv 34 \pmod{89}$.

c) In Exercise 5d we found that an inverse of 89 modulo 232 is 73. Therefore we multiply both sides of this equation by 73, obtaining $x \equiv 146 \pmod{232}$. As a check, we compute $89 \cdot 146 = 12994 \equiv 2 \pmod{232}$.

13. We follow the hint. Adding 6 to both sides gives the equivalent congruence $15x^2 + 19x + 6 \equiv 0 \pmod{11}$, because $5 + 6 = 11 \equiv 0 \pmod{11}$. This factors as $(5x + 3)(3x + 2) \equiv 0 \pmod{11}$. Because there are no non-zero divisors of 0 working modulo 11, we conclude that the solutions are precisely the solutions of $5x + 3 \equiv 0 \pmod{11}$ and $3x + 2 \equiv 0 \pmod{11}$. We solve these by the method of Example 3. By inspection (trial-and-error) or working it out through the Euclidean algorithm and back-substituting, we find that an inverse of 5 modulo 11 is 9, and multiplying both sides of $5x+3 \equiv 0 \pmod{11}$ by 9 yields $x+27 \equiv 0 \pmod{11}$, so $x \equiv -27 \equiv 6 \pmod{11}$. Similarly, an inverse of 3 modulo 11 is 4, and we get $x \equiv -8 \equiv 3 \pmod{11}$. So the solution set is $\{3, 6\}$ (and anything congruent to these modulo 11). Plugging these values into the original equation to check, we have $15 \cdot 3^2 + 19 \cdot 3 + 6 = 198 \equiv 0 \pmod{11}$ and $15 \cdot 6^2 + 19 \cdot 6 + 6 = 660 \equiv 0 \pmod{11}$.

15. The hypothesis tells us that m divides $ac - bc$, which is the product $(a - b)c$. Let m' be $m/\gcd(c, m)$. Then m' is a factor of m, so certainly $m' \mid (a - b)c$. Now since all the common factors of m and c were divided out of m to get m', we know that m' is relatively prime to c. It follows from Lemma 2 in Section 4.3 that $m' \mid a - b$. But this means that $a \equiv b \pmod{m'}$, exactly what we were trying to prove.

17. We want to find numbers x such that $x^2 \equiv 1 \pmod{p}$, in other words, such that p divides $x^2 - 1$. Factoring this expression, we see that we are seeking numbers x such that $p \mid (x+1)(x-1)$. By Lemma 3 in Section 4.3, this can only happen if $p \mid x + 1$ or $p \mid x - 1$. But these two congruences are equivalent to the statements $x \equiv -1 \pmod{p}$ and $x \equiv 1 \pmod{p}$.

19. **a)** If two of these integers were congruent modulo p, say ia and ja, where $1 \leq i < j < p$, then we would have $p \mid ja - ia$, or $p \mid (j - i)a$. By Lemma 2 (or Lemma 3) in Section 4.3, since a is not divisible by p, p must divide $j - i$. But this is impossible, since $j - i$ is a positive integer less than p. Therefore no two of these integers are congruent modulo p.

b) By part **(a)**, since no two of a, $2a$, ..., $(p - 1)a$ are congruent modulo p, each must be congruent to a different number from 1 to $p-1$. Therefore if we multiply them all together, we will obtain the same product, modulo p, as if we had multiplied all the numbers from 1 to $p - 1$. In symbols,

$$a \cdot 2a \cdot 3a \cdots (p-1)a \equiv 1 \cdot 2 \cdot 3 \cdots (p-1) \pmod{p}.$$

The left-hand side of this congruence is clearly $(p - 1)! \cdot a^{p-1}$, and the right-hand side is just $(p - 1)!$, as desired.

c) Wilson's theorem says that $(p - 1)!$ is congruent to -1 modulo p. Therefore the congruence in part **(b)** says that $(-1) \cdot a^{p-1} \equiv -1 \pmod{p}$. Multiplying both sides by -1, we see that $a^{p-1} \equiv 1 \pmod{p}$, as desired. Note that we already assumed the hypothesis that $p \nmid a$ in part **(a)**.

d) If $p \mid a$, then both sides of $a^p \equiv a \pmod{p}$ are 0 modulo p, so the congruence holds. If not, then we just multiply the result obtained in part **(c)** by a.

21. Since 2, 3, 5, and 11 are pairwise relatively prime, we can use the Chinese remainder theorem. The answer will be unique modulo $2 \cdot 3 \cdot 5 \cdot 11 = 330$. Using the notation in the text, we have $a_1 = 1$, $m_1 = 2$, $a_2 = 2$, $m_2 = 3$, $a_3 = 3$, $m_3 = 5$, $a_4 = 4$, $m_4 = 11$, $m = 330$, $M_1 = 330/2 = 165$, $M_2 = 330/3 = 110$, $M_3 = 330/5 = 66$, $M_4 = 330/11 = 30$. Then we need to find inverses y_i of M_i modulo m_i for $i = 1, 2, 3, 4$. This can be done by inspection (trial and error), since the moduli here are so small, or systematically using the Euclidean algorithm, as in Exercise 5; we find that $y_1 = 1$, $y_2 = 2$, $y_3 = 1$, and $y_4 = 7$ (for this last one, $30 \equiv 8 \pmod{11}$, so we want to solve $8y_4 = 1 \pmod{11}$, and we observe that $8 \cdot 7 = 56 \equiv 1 \pmod{11}$). Thus our solution is $x = 1 \cdot 165 \cdot 1 + 2 \cdot 110 \cdot 2 + 3 \cdot 66 \cdot 1 + 4 \cdot 30 \cdot 7 = 1643 \equiv 323 \pmod{330}$. So the solutions are all integers of the form $323 + 330k$, where k is an integer.

23. By definition, the first congruence can be written as $x = 3t + 2$ where t is an integer. Substituting this expression for x into the second congruence tells us that $3t + 2 \equiv 1 \pmod 4$, which can easily be solved to show that $t \equiv 1 \pmod 4$. From this we can write $t = 4u + 1$ for some integer u. Thus $x = 3t + 2 = 3(4u + 1) + 2 = 12u + 5$. We plug this into the third congruence to obtain $12u + 5 \equiv 3 \pmod 5$, which we easily solve to give $u \equiv 4 \pmod 5$. Hence $u = 5v + 4$, and so $x = 12u + 5 = 12(5v + 4) + 5 = 60v + 53$. We check our answer by confirming that $53 \equiv 2 \pmod 3$, $53 \equiv 1 \pmod 4$, and $53 \equiv 3 \pmod 5$.

25. We simply translate the steps of the calculation given in the proof of Theorem 2 into pseudocode. Of course, hidden in line 7 below is a multi-step process of finding inverses in modular arithmetic, which can be accomplished by using the Euclidean algorithm and back-substituting, as in Example 2. The last loop reduces the answer x to its simplest form modulo m. All solutions are then of the form $x + mk$, where m is the product of the moduli and k is an integer.

procedure *chinese*($m_1, m_2, \ldots, m_n$: relatively prime positive integers; $a_1, a_2, \ldots, a_n$: integers)
$m := 1$
for $k := 1$ **to** n
    $m := m \cdot m_k$
for $k := 1$ **to** n
    $M_k := m/m_k$
    $y_k := M_k^{-1}$ **mod** m_k
$x := 0$
for $k := 1$ **to** n
    $x := x + a_k M_k y_k$
while $x \geq m$
    $x := x - m$
return x {the smallest solution to the system $\{ x \equiv a_k \pmod{m_k},\ k = 1, 2, \ldots, n \}$}

27. We cannot apply the Chinese remainder theorem directly, since the moduli are not pairwise relatively prime. However, we can, using the Chinese remainder theorem, translate these congruences into a set of congruences that together are equivalent to the given congruence. Since we want $x \equiv 4 \pmod{12}$, we must have $x \equiv 4 \equiv 1 \pmod 3$ and $x \equiv 4 \equiv 0 \pmod 4$. Similarly, from the third congruence we must have $x \equiv 1 \pmod 3$ and $x \equiv 2 \pmod 7$. Since the first congruence is consistent with the requirement that $x \equiv 1 \pmod 3$, we see that our system is equivalent to the system $x \equiv 7 \pmod 9$, $x \equiv 0 \pmod 4$, $x \equiv 2 \pmod 7$. These can be solved using the Chinese remainder theorem (see Exercise 21 or Example 5) to yield $x \equiv 16 \pmod{252}$. Therefore the solutions are all integers of the form $16 + 252k$, where k is an integer.

29. We will argue for the truth of this statement using the Fundamental Theorem of Arithmetic. What we must show is that $m_1 m_2 \cdots m_n \mid a - b$. Look at the prime factorization of both sides of this proposition. Suppose that p is a prime appearing in the prime factorization of the left-hand side. Then $p \mid m_j$ for some j. Since the m_i's are relatively prime, p does not appear as a factor in any of the other m_i's. Now we know from the hypothesis that $m_j \mid a - b$. Therefore $a - b$ contains the factor p in its prime factorization, and p must appear to a power at least as large as the power to which it appears in m_j. But what we have just shown is that each prime power p^r in the prime factorization of the left-hand side also appears in the prime factorization of the right-hand side. Therefore the left-hand side does, indeed, divide the right-hand side.

31. We are asked to solve the simultaneous congruences $x \equiv 1 \pmod 2$ and $x \equiv 1 \pmod 3$. The solution will be unique modulo $2 \cdot 3 = 6$. By inspection we see that the answer is simply that $x \equiv 1 \pmod 6$. The solution set is $\{\ldots, -11, -5, 1, 7, 13, \ldots\}$.

33. Fermat's little theorem tells us that $7^{12} \equiv 1 \pmod{13}$. Note that $121 = 10 \cdot 12 + 1$. Therefore $7^{121} = 7^{12 \cdot 10} \cdot 7 = (7^{12})^{10} \cdot 7 \equiv 1^{10} \cdot 7 = 7 \pmod{13}$.

35. Fermat's little theorem tells us that under the given conditions $a^{p-1} \equiv 1 \pmod{p}$. Therefore $a^{p-2} \cdot a = a \cdot a^{p-2} = a^{p-1} \equiv 1 \pmod{p}$. This is precisely the definition that a^{p-2} is an inverse of a modulo p.

37. **a)** We calculate $2^{340} = (2^{10})^{34} \equiv 1^{34} = 1 \pmod{11}$, since Fermat's little theorem says that $2^{10} \equiv 1 \pmod{11}$.

b) We calculate $2^{340} = (2^5)^{68} = 32^{68} \equiv 1^{68} = 1 \pmod{31}$, since $32 \equiv 1 \pmod{31}$.

c) Since 11 and 31 are relatively prime, and $11 \cdot 31 = 341$, it follows from the first two parts and Exercise 29 that $2^{340} \equiv 1 \pmod{341}$.

39. **a)** By Fermat's little theorem we know that $5^6 \equiv 1 \pmod{7}$; therefore $5^{1998} = (5^6)^{333} \equiv 1^{333} \equiv 1 \pmod{7}$, and so $5^{2003} = 5^5 \cdot 5^{1998} \equiv 3125 \cdot 1 \equiv 3 \pmod{7}$. So 5^{2003} **mod** $7 = 3$. Similarly, $5^{10} \equiv 1 \pmod{11}$; therefore $5^{2000} = (5^{10})^{200} \equiv 1^{200} \equiv 1 \pmod{11}$, and so $5^{2003} = 5^3 \cdot 5^{2000} \equiv 125 \cdot 1 \equiv 4 \pmod{11}$. So 5^{2003} **mod** $11 = 4$. Finally, $5^{12} \equiv 1 \pmod{13}$; therefore $5^{1992} = (5^{12})^{166} \equiv 1^{166} \equiv 1 \pmod{13}$, and so $5^{2003} = 5^{11} \cdot 5^{1992} \equiv 48{,}828{,}125 \cdot 1 \equiv 8 \pmod{13}$. So 5^{2003} **mod** $13 = 8$.

b) We now apply the Chinese remainder theorem to the results of part **(a)**, as in Example 5. Let $m = 7 \cdot 11 \cdot 13 = 1001$, $M_1 = m/7 = 143$, $M_2 = m/11 = 91$, and $M_3 = m/13 = 77$. We see that 5 is an inverse of 143 modulo 7, since $143 \equiv 3 \pmod{7}$, and $3 \cdot 5 = 15 \equiv 1 \pmod{7}$. Similarly, 4 is an inverse of 91 modulo 11, and 12 is an inverse of 77 modulo 13. (An algorithm to compute inverses—if we don't want to find them by inspection as we've done here—is illustrated in Example 2.) Therefore the answer is $(3 \cdot 143 \cdot 5 + 4 \cdot 91 \cdot 4 + 8 \cdot 77 \cdot 12)$ **mod** $1001 = 10993$ **mod** $1001 = 983$.

41. Let q be a (necessarily odd) prime dividing $2^p - 1$. By Fermat's little theorem, we know that $q \mid 2^{q-1} - 1$. Then from Exercise 37 in Section 4.3 we know that $\gcd(2^p - 1, 2^{q-1} - 1) = 2^{\gcd(p,q-1)} - 1$. Since q is a common divisor of $2^p - 1$ and $2^{q-1} - 1$, we know that $\gcd(2^p - 1, 2^{q-1} - 1) > 1$. Hence $\gcd(p, q-1) = p$, since the only other possibility, namely $\gcd(p, q-1) = 1$, would give us $\gcd(2^p - 1, 2^{q-1} - 1) = 1$. Hence $p \mid q - 1$, and therefore there is a positive integer m such that $q - 1 = mp$. Since q is odd, m must be even, say $m = 2k$, and so every prime divisor of $2^p - 1$ is of the form $2kp + 1$. Furthermore, products of numbers of this form are also of this form, since $(2k_1p + 1)(2k_2p + 1) = 4k_1k_2p^2 + 2k_1p + 2k_2p + 1 = 2(2k_1k_2p + k_1 + k_2)p + 1$. Therefore all divisors of $2^p - 1$ are of this form.

43. To decide whether $2^{11} - 1 = 2047$ is prime, we need only look for a prime factor not exceeding $\sqrt{2047} \approx 45$. By Exercise 41 every such prime divisor must be of the form $22k + 1$. The only candidate is therefore 23. In fact $2047 = 23 \cdot 89$, so we conclude that 2047 is not prime.

We can take the same approach for $2^{17} - 1 = 131{,}071$, but we need either computer algebra software or patience with a calculator. By Exercise 41 every prime divisor of $2^{17} - 1$ must be of the form $34k + 1$, so we need to try all such divisors (or at least those that are not obviously nonprime) up to $\sqrt{131{,}071} \approx 362$, which means up to $k = 10$. No number of this form divides 131,071, so we conclude that it is prime.

45. First note that $2047 = 23 \cdot 89$, so 2047 is composite. To apply Miller's test, we write $2047 - 1 = 2046 = 2 \cdot 1023$, so $s = 1$ and $t = 1023$. We must show that either $2^{1023} \equiv 1 \pmod{2047}$ or $2^{1023} \equiv -1 \pmod{2047}$. To compute, we write $2^{1023} = (2^{11})^{93} = 2048^{93} \equiv 1^{93} = 1 \pmod{2047}$, as desired. (We could also compute this using the modular exponentiation algorithm given in Section 4.2—see Example 12 in that section.)

47. We factor $2821 = 7 \cdot 13 \cdot 31$. We must show that this number meets the definition of Carmichael number, namely that $b^{2820} \equiv 1 \pmod{2821}$ for all b relatively prime to 2821. Note that if $\gcd(b, 2821) = 1$, then $\gcd(b, 7) = \gcd(b, 13) = \gcd(b, 31) = 1$. Using Fermat's little theorem we find that $b^6 \equiv 1 \pmod{7}$, $b^{12} \equiv 1 \pmod{13}$, and $b^{30} \equiv 1 \pmod{31}$. It follows that $b^{2820} = (b^6)^{470} \equiv 1 \pmod{7}$, $b^{2820} = (b^{12})^{235} \equiv 1 \pmod{13}$, and $b^{2820} = (b^{30})^{94} \equiv 1 \pmod{31}$. By Exercise 29 (or the Chinese remainder theorem) it follows that $b^{2820} \equiv 1 \pmod{2821}$, as desired.

49. **a)** If we multiply out this expression, we get $n = 1296m^3 + 396m^2 + 36m + 1$. Clearly $6m \mid n-1$, $12m \mid n-1$, and $18m \mid n-1$. Therefore, the conditions of Exercise 48 are met, and we conclude that n is a Carmichael number.

b) Letting $m = 51$ gives $n = 172{,}947{,}529$. We note that $6m+1 = 307$, $12m+1 = 613$, and $18m+1 = 919$ are all prime.

51. It is straightforward to calculate the remainders when the integers from 0 to 14 are divided by 3 and by 5. For example, the remainders when 10 is divided by 3 and 5 are 1 and 0, respectively, so we represent 10 by the pair $(1,0)$. The exercise is simply asking us to tabulate these remainders, as in Example 7.

$0 = (0,0)$	$3 = (0,3)$	$6 = (0,1)$	$9 = (0,4)$	$12 = (0,2)$
$1 = (1,1)$	$4 = (1,4)$	$7 = (1,2)$	$10 = (1,0)$	$13 = (1,3)$
$2 = (2,2)$	$5 = (2,0)$	$8 = (2,3)$	$11 = (2,1)$	$14 = (2,4)$

53. The method of solving a system of congruences such as this is given in the proof of Theorem 2. Here we have $m_1 = 99$, $m_2 = 98$, $m_3 = 97$, and $m_4 = 95$, so that $m = 99 \cdot 98 \cdot 97 \cdot 95 = 89403930$. We compute the values $M_k = m/m_k$ and obtain $M_1 = 903070$, $M_2 = 912285$, $M_3 = 921690$, and $M_4 = 941094$. Next we need to find the inverses y_k of M_k modulo m_k. To do this we first replace each M_k by its remainder modulo m_k (to make the arithmetic easier), and then apply the technique shown in Example 2. For $k = 1$ we want to find the inverse of 903070 modulo 99, which is the same as the inverse of 903070 **mod** 99, namely 91. To do this we apply the Euclidean algorithm to express 1 as a linear combination of 91 and 99.

$$\begin{aligned} 99 &= 91 + 8 \\ 91 &= 11 \cdot 8 + 3 \\ 8 &= 2 \cdot 3 + 2 \\ 3 &= 2 + 1 \\ \therefore\ 1 &= 3 - 2 \\ &= 3 - (8 - 2 \cdot 3) = 3 \cdot 3 - 8 \\ &= 3 \cdot (91 - 11 \cdot 8) - 8 = 3 \cdot 91 - 34 \cdot 8 \\ &= 3 \cdot 91 - 34 \cdot (99 - 91) = 37 \cdot 91 - 34 \cdot 99 \end{aligned}$$

We therefore conclude that the inverse of 91 modulo 99 is 37, so we have $y_1 = 37$. Similar calculations show that $y_2 = 33$, $y_3 = 24$, and $y_4 = 4$. Continuing with the procedure outlined in the proof of Theorem 2, we now form the sum of the products $a_k M_k y_k$, and this will be our solution. We have

$$65 \cdot 903070 \cdot 37 + 2 \cdot 912285 \cdot 33 + 51 \cdot 921690 \cdot 24 + 10 \cdot 941094 \cdot 4 = 3397886480.$$

We want our answer reduced modulo m, so we divide by 89403930 and take the remainder, obtaining 537140. (All of these calculations are not difficult using a scientific calculator.) Finally, let us check our answer: 537140 **mod** 99 = 65, 537140 **mod** 98 = 2, 537140 **mod** 97 = 51, 537140 **mod** 95 = 10.

55. For the first question we seek an exponent n such that $2^n \equiv 5 \pmod{19}$. For the second we want $2^n \equiv 6 \pmod{19}$. There is no known efficient algorithm for finding these exponents, so we might as well just start computing powers of 2 modulo 19. In each case, we just need to multiply the previous result by 2, working modulo 19. We have $2^2 = 4 \pmod{19}$, $2^3 = 2 \cdot 4 = 8 \pmod{19}$, $2^4 = 2 \cdot 8 = 16 \pmod{19}$, $2^5 = 2 \cdot 16 = 32 \equiv 13 \pmod{19}$, $2^6 = 2 \cdot 2^5 \equiv 2 \cdot 13 = 26 \equiv 7 \pmod{19}$, $2^7 \equiv 2 \cdot 7 = 14 \pmod{19}$, $2^8 \equiv 2 \cdot 14 = 28 \equiv 9 \pmod{19}$, $2^9 \equiv 2 \cdot 9 = 18 \pmod{19}$, $2^{10} \equiv 2 \cdot 18 = 36 \equiv 17 \pmod{19}$, $2^{11} \equiv 2 \cdot 17 = 34 \equiv 15 \pmod{19}$, $2^{12} \equiv 2 \cdot 15 = 30 \equiv 11 \pmod{19}$, $2^{13} \equiv 2 \cdot 11 = 22 \equiv 3 \pmod{19}$, $2^{14} \equiv 2 \cdot 3 = 6 \pmod{19}$. Finally! So we conclude that the discrete logarithm of 6 to the base 2 modulo 19 is 14. Continuing the calculation, we have $2^{15} \equiv 2 \cdot 6 = 12 \pmod{19}$, $2^{16} \equiv 2 \cdot 12 = 24 \equiv 5 \pmod{19}$. So the discrete logarithm of 5 to the base 2 modulo 19 is 16.

57. A computer algebra system such as *Maple* facilitates the modular arithmetic calculations. We repeatedly multiply by 3 and reduce modulo 17. We get $3^0 = 1 \pmod{17}$, $3^1 = 3 \pmod{17}$, $3^2 = 9 \pmod{17}$, $3^3 = 27 \equiv 10 \pmod{17}$, and so on. Thus $\log_3 1 = 0$, $\log_3 3 = 1$, $\log_3 9 = 2$, $\log_3 10 = 3$, and so on. If we collect the data and present them in order of increasing argument, we get the required table. (Of course $\log_3 0$ does not exist.)

$$\log_3 1 = 0 \quad \log_3 2 = 14 \quad \log_3 3 = 1 \quad \log_3 4 = 12 \quad \log_3 5 = 5 \quad \log_3 6 = 15 \quad \log_3 7 = 11 \quad \log_3 8 = 10$$

$$\log_3 9 = 2 \quad \log_3 10 = 3 \quad \log_3 11 = 7 \quad \log_3 12 = 13 \quad \log_3 13 = 4 \quad \log_3 14 = 9 \quad \log_3 15 = 6 \quad \log_3 16 = 8$$

59. We need to prove that if the congruence $x^2 \equiv a \pmod{p}$ has any solutions at all, then it has exactly two solutions. So let us assume that s is a solution. Clearly $-s$ is a solution as well, since $(-s)^2 = s^2$. Furthermore, $-s \not\equiv s \pmod{p}$, since if it were, we would have $2s \equiv 0 \pmod{p}$, which means that $p \mid 2s$. Since p is an odd prime, that means that $p \mid s$, so that $s \equiv 0 \pmod{p}$. Therefore $a \equiv 0 \pmod{p}$, contradicting the conditions of the problem.

It remains to prove that there cannot be more than two incongruent solutions. Suppose that s is one solution and that t is a second solution. We have $s^2 \equiv t^2 \pmod{p}$. This means that $p \mid s^2 - t^2$, that is, $p \mid (s+t)(s-t)$. Since p is prime, Lemma 3 in Section 4.3 guarantees that $p \mid s-t$ or $p \mid s+t$. This means that $t \equiv s \pmod{p}$ or $t \equiv -s \pmod{p}$. Therefore any solution t must be either the first solution or its negative. In other words, there are at most two solutions.

61. There is really almost nothing to prove here. The value $\left(\frac{a}{p}\right)$ depends only on whether or not a is a quadratic residue modulo p, i.e., whether or not the equivalence $x^2 \equiv a \pmod{p}$ has a solution. Obviously, this depends only on the equivalence class of a modulo p.

63. By Exercise 62 we know that $\left(\frac{a}{p}\right)\left(\frac{b}{p}\right) \equiv a^{(p-1)/2} b^{(p-1)/2} = (ab)^{(p-1)/2} \equiv \left(\frac{ab}{p}\right) \pmod{p}$. Since the only values either side of this equivalence can take on are ± 1, being congruent modulo p is the same as being equal.

65. We follow the hint. Working modulo 5, we want to solve $x^2 \equiv 4$. It is easy to see that there are exactly two solutions modulo 5, namely $x = 2$ and $x = 3$. Similarly there are only the solutions $x = 1$ and $x = 6$ modulo 7. Therefore we want to find values of x modulo $5 \cdot 7 = 35$ such that $x \equiv 2$ or 3 (mod 5) and $x \equiv 1$ or 6 (mod 7). We can do this by applying the Chinese remainder theorem (as in Example 5) four times, for the four combinations of these values. For example, to solve $x \equiv 2 \pmod{5}$ and $x \equiv 1 \pmod{7}$, we find that $m = 35$, $M_1 = 7$, $M_2 = 5$, $y_1 = 3$, $y_2 = 3$, so $x \equiv 2 \cdot 7 \cdot 3 + 1 \cdot 5 \cdot 3 = 57 \equiv 22 \pmod{35}$. Doing the similar calculation with the other three possibilities yields the other three solutions modulo 35: $x = 8$, $x = 13$, and $x = 27$.

67. To compute $\log_r a \pmod{p}$, we need to solve $r^e \equiv a \pmod{p}$ for e. The brute force approach is just to compute r^e **mod** p for $e = 0, 1, 2, \ldots, p-2$ until we get the answer a. This requires about p iterations, each of which can be done with $O(\log p)$ bit operations, since we need only multiply the previous value by r and find the remainder upon division by p. At worst, we require all p iterations; on average, only half that many. In either case, the time complexity is $O(p \log p)$, which is prohibitively large if p is, say, a 200-digit number.

SECTION 4.5 Applications of Congruences

The great British number theorist G. H. Hardy (1877–1947) once said, "I have never done anything 'useful.' No discovery of mine has made, or is likely to make, directly or indirectly, for good or ill, the least difference to the amenity of the world." He was wrong. Number theory has many applications, especially in cryptography (see Section 4.6). In the present section we saw applications to hashing functions (important for storing large amounts of information and being able to retrieve it efficiently), pseudorandom numbers (important for computer simulations), and check digits (important in our technological world). Hardy would be appalled! The exercises in this section are mostly routine.

1. We are simply asked to compute k **mod** 97 for each value of k. We do this by dividing the given number by 97 and taking the remainder, which can be found either by multiplying the decimal remainder by 97, or by subtracting 97 times the quotient from k. (See the solution to Exercise 3 below for details.)

a) 034567981 **mod** 97 = 91 **b)** 183211232 **mod** 97 = 57

c) 220195744 **mod** 97 = 21 **d)** 987255335 **mod** 97 = 5

3. a) We need to compute k **mod** 31 in each case. A good way to do this on a calculator is as follows. Enter k and divide by 31. The result will be a number with an integer part and a decimal fractional part. Subtract off the integer part, leaving a decimal fraction between 0 and 1. This is the remainder expressed as a decimal. To find out what whole number remainder that really represents, multiply by 31. The answer will be a whole number (or nearly so—it may require rounding, say from 4.9999 or 5.0001 to 5), and that number is k **mod** 31.

(i) 317 **mod** 31 = 7 (ii) 918 **mod** 31 = 19 (iii) 007 **mod** 31 = 7

(iv) 100 **mod** 31 = 7 (v) 111 **mod** 31 = 18 (vi) 310 **mod** 31 = 0

b) Take the next available space, where the next space is computed by adding 1 to the space number and pretending that $30 + 1 = 0$.

5. We apply the formula with $n = 0$ to obtain $x_1 = (3 \cdot x_0 + 2)$ **mod** 13 $= (3 \cdot 1 + 2)$ **mod** 13 $= 5$. Then $x_2 = (3 \cdot x_1 + 2)$ **mod** 13 $= (3 \cdot 5 + 2)$ **mod** 13 $= 17$ **mod** 13 $= 4$. Continuing in this way we have $x_3 = (3 \cdot 4 + 2)$ **mod** 13 $= 1$. Because this is the same as x_0, the sequence repeats from here on out. So the sequence is $1, 5, 4, 1, 5, 4, 1, 5, 4, \ldots$.

7. We compute until the sequence begins to repeat:

$$x_1 = 3 \cdot 2 \textbf{ mod } 11 = 6$$
$$x_2 = 3 \cdot 6 \textbf{ mod } 11 = 7$$
$$x_3 = 3 \cdot 7 \textbf{ mod } 11 = 10$$
$$x_4 = 3 \cdot 10 \textbf{ mod } 11 = 8$$
$$x_5 = 3 \cdot 8 \textbf{ mod } 11 = 2$$

Since $x_5 = x_0$, the sequence repeats forever: $2, 6, 7, 10, 8, 2, 6, 7, 10, 8, \ldots$.

9. We follow the instructions. Because $2357^2 = 5555449 = 05555449$, the middle four digits are 5554, so 5554 is our second pseudorandom number. Next $5554^2 = 30846916$, so our third pseudorandom number is 8469. Repeating the same procedure leads to the following five terms: $7239, 4031, 2489, 1951, 8064$.

11. We are told to apply the formula $x_{n+1} = x_n^3$ **mod** 7, starting with $x_0 = 2$. Thus $x_1 = 2^3$ **mod** 7 $= 1$, $x_2 = 1^3$ **mod** 7 $= 1$, and our sequence never gets off the ground! The sequence generated here is $2, 1, 1, 1, \ldots$.

13. A correctly transmitted bit string must have an even number of 1's. Therefore we can be sure that there is an error in (d), but because the other three strings have an even number of 1's, we cannot detect an error in any of them. (Of course that doesn't mean that there is no error, because it is possible that two bits were transmitted incorrectly, in which case the sum modulo 2 does not change.)

15. Let d be the check digit. Then we know that $1{\cdot}0{+}2{\cdot}0{+}3{\cdot}7{+}4{\cdot}1{+}5{\cdot}1{+}6{\cdot}9{+}7{\cdot}8{+}8{\cdot}8{+}9{\cdot}1{+}10{\cdot}d \equiv 0 \pmod{11}$. This simplifies to $213{+}10{\cdot}d \equiv 0 \pmod{11}$. But $213 \equiv 4 \pmod{11}$, and $10 \equiv -1 \pmod{11}$, so this is equivalent to $4 - d \equiv 0 \pmod{11}$, or $d = 4$.

17. The ISBN is 0073383090. To check its validity we compute, as in Example 6, $1 \cdot 0 + 2 \cdot 0 + 3 \cdot 7 + 4 \cdot 3 + 5 \cdot 3 + 6 \cdot 8 + 7 \cdot 3 + 8 \cdot 0 + 9 \cdot 9 + 10 \cdot 0 = 198$. Because this is congruent to 0 modulo 11, the check digit was computed correctly.

19. To determine whether an 11-digit number is a valid USPS money order identification number, we need to verify that the sum of the first ten digits reduced modulo 9 gives the last digit.

a) $7 + 4 + 0 + 5 + 1 + 4 + 8 + 9 + 6 + 2 \bmod 9 = 46 \bmod 9 = 1 \neq 3$, so this is not a valid number.

b) $8 + 8 + 3 + 8 + 2 + 0 + 1 + 3 + 4 + 4 \bmod 9 = 41 \bmod 9 = 5$, which is the last digit, so this is a valid number.

c) $5 + 6 + 1 + 5 + 2 + 2 + 4 + 0 + 7 + 8 \bmod 9 = 40 \bmod 9 = 4$, which is the last digit, so this is a valid number.

d) $6 + 6 + 6 + 0 + 6 + 6 + 3 + 1 + 1 + 7 \bmod 9 = 42 \bmod 9 = 6 \neq 8$, so this is not a valid number.

21. In each case, we know that $x_{11} = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} \bmod 9$. (See the preamble to Exercise 18.) This is equivalent to $x_{11} \equiv x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} \pmod{9}$, with $0 \leq x_{11} \leq 8$. Therefore we will get an equation modulo 9 involving the unknown Q for each of these valid postal money order identification numbers.

a) $8 \equiv 4 + 9 + 3 + 2 + 1 + 2 + Q + 0 + 6 + 8 \pmod{9}$, which is equivalent to $8 \equiv Q + 35 \equiv Q + 8 \pmod{9}$. Therefore $Q \equiv 0 \pmod{9}$. There are two single-digit numbers Q that makes this true: $Q = 0$ and $Q = 9$, so it is impossible to know for sure what the smudged digit was.

b) $8 \equiv 8 + 5 + 0 + Q + 9 + 1 + 0 + 3 + 8 + 5 \pmod{9}$, which is equivalent to $8 \equiv Q + 39 \equiv Q + 3 \pmod{9}$. The only single-digit number Q that makes this true is $Q = 5$, so the smudged digit must have been a 5.

c) $4 \equiv 2 + Q + 9 + 4 + 1 + 0 + 0 + 7 + 7 + 3 \pmod{9}$, which is equivalent to $4 \equiv Q + 33 \equiv Q + 6 \pmod{9}$. The only single-digit number Q that makes this true is $Q = 7$, so the smudged digit must have been a 7.

d) $1 \equiv 6 + 6 + 6 + 8 + 7 + Q + 0 + 3 + 2 + 0 \pmod{9}$, which is equivalent to $1 \equiv Q + 38 \equiv Q + 2 \pmod{9}$. The only single-digit number Q that makes this true is $Q = 8$, so the smudged digit must have been an 8.

23. Because the first ten digits are added, any transposition error involving them will go undetected—the sum of the first ten digits will be the same for the transposed number as it is for the correct number. Suppose the last digit is transposed with another digit; without loss of generality, we can assume it's the tenth digit and that $x_{10} \neq x_{11}$. Then the correct equation will be

$$x_{11} \equiv x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} \pmod{9}$$

but the equation resulting from the error will read

$$x_{10} \equiv x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{11} \pmod{9}.$$

Subtracting these two equations, we see that the erroneous equation will be true if and only if $x_{11} - x_{10} \equiv x_{10} - x_{11} \pmod{9}$. This is equivalent to $2x_{11} \equiv 2x_{10} \pmod{9}$, which, because 2 is relatively prime to 9, is equivalent to $x_{11} \equiv x_{10} \pmod{9}$, which is false. This tells us that the check equation will fail. Therefore we conclude that transposition errors involving the eleventh digits are detected.

25. From Example 5. we know that a valid UPC code must satisfy the equation

$$3x_1 + x_2 + 3x_3 + x_4 + 3x_5 + x_6 + 3x_7 + x_8 + 3x_9 + x_{10} + 3x_{11} + x_{12} \equiv 0 \pmod{10}.$$

Therefore in each case we simply need to compute the left-hand side of this equation modulo 10 and see whether or not we get 0 as the answer.

a) $3 \cdot 0 + 3 + 3 \cdot 6 + 0 + 3 \cdot 0 + 0 + 3 \cdot 2 + 9 + 3 \cdot 1 + 4 + 3 \cdot 5 + 2 = 60 \equiv 0 \pmod{10}$, so this is a valid code.

b) $3 \cdot 0 + 1 + 3 \cdot 2 + 3 + 3 \cdot 4 + 5 + 3 \cdot 6 + 7 + 3 \cdot 8 + 9 + 3 \cdot 0 + 3 = 88 \not\equiv 0 \pmod{10}$, so this is not a valid code.

c) $3 \cdot 7 + 8 + 3 \cdot 2 + 4 + 3 \cdot 2 + 1 + 3 \cdot 8 + 4 + 3 \cdot 3 + 0 + 3 \cdot 1 + 4 = 90 \equiv 0 \pmod{10}$, so this is a valid code.

d) $3 \cdot 7 + 2 + 3 \cdot 6 + 4 + 3 \cdot 1 + 2 + 3 \cdot 1 + 7 + 3 \cdot 5 + 4 + 3 \cdot 2 + 5 = 90 \equiv 0 \pmod{10}$, so this is a valid code.

27. The digits with even subscripts appear in the formula with coefficient 1, whereas those with odd subscripts appear with coefficient 3. Therefore if two digits whose positions have the same parity (both odd or both even) are switched, then the sum will be unchanged and such an error cannot be detected. If two digits whose parities are different are transposed, say an x in an odd position and a y in an even position, then the new sum will differ from the old sum by $(x + 3y) - (3x + y)$, which equals $2(y - x)$. As long as the two transposed digits do not differ by 5, the sum will therefore be different modulo 10; if they do differ by 5, then the sum will be the same modulo 10. We conclude that transposition errors will be detected if and only if the transposed digits are an odd number of positions apart (in particular transposing neighboring digits) and do not differ by 5.

29. In each case we need to compute $a_1 a_2 \ldots a_{14}$ **mod** 7 and see if we get a_{15}. This may be inconvenient on a calculator with only 12 digits of precision, but one can always divide it out by hand (or, better, use computer algebra software).

a) $10133334178901 = 7 \cdot 1447619168414 + 3$. Therefore 10133334178901 **mod** $7 = 3 = a_{15}$, so this is a valid airline ticket number. (In *Maple* we could just type `10133334178901 mod 7` and get the response `3`.)

b) 00786234277044 **mod** $7 = 6 \neq 5 = a_{15}$, so this is not a valid number.

c) 11327343888253 **mod** $7 = 1 = a_{15}$, so this is a valid number.

d) 00012234732287 **mod** $7 = 1 = a_{15}$, so this is a valid number.

31. Let's solve a more general problem by ignoring the word "consecutive." First we look at the case in which the transposition does not involve the check digit itself. Suppose the erroneous number formed by the first 14 digits occurs when a_i is interchanged with a_j, where $1 \leq i < j \leq 14$. Because of our decimal place-value numeration system, before the switch, a_i was contributing $a_i \cdot 10^{14-i}$ to the value of the number, and a_j was contributing $a_j \cdot 10^{14-j}$. Therefore this change has increased the 14-digit number by $(a_j - a_i)10^{14-i} + (a_i - a_j)10^{14-j}$, which equals $(a_j - a_i)(10^{14-i} - 10^{14-j})$. In order for this to still check, this last expression must be equivalent to 0 modulo 7. Obviously this will happen if a_i and a_j differ by 7, but it will also happen if $(10^{14-i} - 10^{14-j})$ is a multiple of 7. A bit of calculation shows that this will happen if and only if $j - i = 6$ or 12. Thus we cannot detect the error if the columns in which the transposition occurs are 6 or 12 apart or the transposed digits differ by 7. Finally, if the digit a_{15} is transposed with the digit a_i, where $1 \leq i \leq 14$, then $a_1 a_2 \ldots a_{14}$ **mod** 7 has gone up by $(a_{15} - a_i)10^{14-i}$ and the check digit has gone up by $a_i - a_{15}$, so we will not be able to detect this error if and only if $(a_{15} - a_i)10^{14-i} \equiv a_i - a_{15} \pmod{7}$, which is equivalent to $(a_{15} - a_i)(10^{14-i} + 1) \equiv 0 \pmod{7}$. Because $10^{14-i} + 1 \equiv 0 \pmod{7}$ if and only if $i = 5$ or 11, we conclude that we cannot detect the transposition error if it interchanges the check digit with a_5 or a_{11} or interchanges it with a digit differing from it by 7. (Of course, the check digit must be 0 through 6, so an error that puts a 7, 8, or 9 in the last position can also be detected.)

Because transposing consecutive digits is not transposing digits whose positions differ by the quantities mentioned above, we can detect all transposition errors of consecutive digits unless the digits differ by 7.

33. In each case we will compute $3d_1 + 4d_2 + 5d_3 + 6d_4 + 7d_5 + 8d_6 + 9d_7$ **mod** 11. If this matches the digit given for d_8, then the ISSN is valid, and conversely.

a) $3 \cdot 1 + 4 \cdot 0 + 5 \cdot 5 + 6 \cdot 9 + 7 \cdot 1 + 8 \cdot 0 + 9 \cdot 2 \bmod 11 = 107 \bmod 11 = 8$. Because $d_8 = 7 \neq 8$, this number is not valid.

b) $3 \cdot 0 + 4 \cdot 0 + 5 \cdot 0 + 6 \cdot 2 + 7 \cdot 9 + 8 \cdot 8 + 9 \cdot 9 \bmod 11 = 220 \bmod 11 = 0$. Because $d_8 = 0$, this number is valid.

c) $3 \cdot 1 + 4 \cdot 5 + 5 \cdot 3 + 6 \cdot 0 + 7 \cdot 8 + 8 \cdot 6 + 9 \cdot 6 \bmod 11 = 196 \bmod 11 = 9$. Because $d_8 = 9$, this number is valid.

d) $3 \cdot 1 + 4 \cdot 0 + 5 \cdot 0 + 6 \cdot 7 + 7 \cdot 1 + 8 \cdot 2 + 9 \cdot 0 \bmod 11 = 68 \bmod 11 = 2$. Because d_8 is "X" (representing 10 modulo 11), this number is not valid.

35. By subtracting d_8 from both sides and noting that $-1 \equiv 10 \pmod{11}$, we see that the checking congruence is equivalent to $3d_1 + 4d_2 + 5d_3 + 6d_4 + 7d_5 + 8d_6 + 9d_7 + 10d_8 \equiv 0 \pmod{11}$. It is now easy to see that transposing adjacent digits x and y (where x is on the left) causes the left-hand side to increase by x and decrease by y, for a net change of $x - y$. Because $x \not\equiv y \pmod{11}$, the congruence will no longer hold. Therefore errors of this type are always detected.

SECTION 4.6 Cryptography

In addition to exercises about the topics covered in this section, this exercise set introduces the Vigenère cipher (Exercises 18–22) and a protocol for key exchange (Exercise 33). There is a nice website for encoding and decoding with the affine cipher (for any function of the form $f(p) = ap + b$), which you can use to check your answers: `www.shodor.org/interactivate/activities/CaesarCipher/`. *A website for the Vigenère cipher can be found here:* `islab.oregonstate.edu/koc/ece575/02Project/Mun+Lee/VigenereCipher.html`

1. a) We need to replace each letter by the letter three places later in the alphabet. Thus D becomes G, O becomes R, and so on. The resulting message is GR QRW SDVV JR.

b) We need to replace each letter by the letter 13 places later in the alphabet. Thus D becomes Q, O becomes B (we cycle, with A following Z), and so on. The resulting message is QB ABG CNFF TB.

c) This one is a little harder, so it is probably easiest to work with the numbers. For D we have $p = 3$ because D is the fourth letter of the alphabet. Then $3 \cdot 3 + 7 \bmod 26 = 16$, so the encrypted letter is the 17^{th} letter, or Q (remember that we start the sequence at 0). Our original message has the following numerical values: 3-14 13-14-19 15-0-18-18 6-14. Multiplying each of these numbers by 3, adding 7, and reducing modulo 26 gives us 16-23 20-23-12 0-7-9-9 25-23. Translating back into letters we have QX UXM AHJJ ZX.

3. In each case, we translate the letters to numbers from 0 to 25, then apply the function, then translate back. (See the solution for Exercise 1c above for details.) In each case, the numerical message is 22-0-19-2-7 24-14-20-17 18-19-4-15.

a) Adding 14 to each number modulo 26 yields 10-14-7-16-21 12-2-8-5 6-7-18-3. Translating back into letters yields KOHQV MCIF GHSD.

b) Multiplying each number by 14, adding 21, and reducing modulo 26 yields 17-21-1-23-15 19-9-15-25 13-1-25-23. Translating back into letters yields RVBXP TJPZ NBZX.

c) Multiplying each number by -7, adding 1, and reducing modulo 26 yields 3-1-24-13-4 15-7-17-12 5-24-25-0. Translating back into letters yields DBYNE PHRM FYZA.

5. a) We need to undo the encryption operation, which was to choose the letter that occurred ten places later in the alphabet. Therefore we need to go backwards 10 places (or, what amounts to the same thing, forward 16 places). For example, the C decodes as S. Doing this for each letter, as in Exercise 1, gives us SURRENDER NOW.

b) BE MY FRIEND **c)** TIME FOR FUN

7. We need to play detective. First note that the two-letter word DY occurs twice. Because this was a shift cipher, we know that the first letter of this word occurs five places beyond the second letter in the alphabet. One of those letters has to be a vowel. This makes it very likely that the word is either UP or TO, corresponding to $k = 9$ or $k = 10$, respectively. Since TO is a more common word, let us assume $k = 10$. To decrypt, we shift each letter in the encrypted message backward 10 places (or forward 16 places) in the alphabet, obtaining TO SLEEP PERCHANCE TO DREAM (from *Hamlet*).

9. Following the same strategy as in Exercise 7, we try to figure out k from the fact that MW is a two-letter word in the encrypted text. What fits best is IS, with $k = 4$. If we apply that to the three-letter word, we get ANY, which seems quite promising. We now decode the entire message: ANY SUFFICIENTLY ADVANCED TECHNOLOGY IS INDISTINGUISHABLE FROM MAGIC.

11. We want to solve the congruence $c \equiv 15p + 13 \pmod{26}$ for p. To do that we will need an inverse of 15 modulo 26, which we can obtain using the Euclidean algorithm or by trial and error. It is 7, because $7 \cdot 15 = 105 = 4 \cdot 26 + 1$. Therefore we have $p = 7(c - 13)$ **mod** $26 = 7c - 91$ **mod** $26 = 7c + 13$ **mod** 26.

13. Because the most common letters are E and T, in that order, and the numerical values of E, T, Z, and J are 4, 19, 25, and 9, respectively, we will assume that $f(4) \equiv 25$ and $f(19) \equiv 9$. This means that $4a + b \equiv 25$ and $19a + b \equiv 9$, where we work modulo 26, of course. Subtracting the two equations gives $15a \equiv 10$, which simplifies to $3a \equiv 2$ (because 5 is not a factor of 26, we can divide both sides by 5). We can find an inverse of 3 modulo 26 using the Euclidean algorithm or by trial and error. It is 9, because $3 \cdot 9 = 27 = 26 + 1$. Therefore $a \equiv 9 \cdot 2 = 18$. Plugging this into $4a + b \equiv 25$ yields $b \equiv 25 - 4a = 25 - 72 \equiv 5$. We therefore guess that the encryption function is $f(p) = 18p + 5$ **mod** 26. As a check, we see that $f(4) = 25$ and $f(19) = 9$.

15. We permute each block of four by undoing the permutation σ. Because $\sigma(1) = 3$, we put the third letter first; because $\sigma(2) = 1$, we put the first letter second; and so on. This gives us BEWA REOF MART IANS, presumably meant to be BEWARE OF MARTIANS.

17. Presumably the message was translated letter by letter, such as by a shift cipher or affine cipher. (Other, nonlinear, bijections on $\mathbf{Z}_{26}$ are also possible.)

19. The numerical version of the encrypted text is 14-8-10-24-22-21-7-1-23. If we subtract the values for the key HOTHOTHOT, namely 7-14-19-7-14-19-7-14-19 and reduce modulo 26, we obtain 7-20-17-17-8-2-0-13-4, which translates to HURRICANE.

21. If l is the distance between the beginnings of the string that occurs several times, then it may be likely that the length of the key string is a factor of l. Thus if we have several such values of l, we can find their greatest common divisor and assume that the length of the key string is a factor of this gcd.

23. Suppose that we know $n = pq$ and $(p - 1)(q - 1)$, and we wish to find p and q. Here is how we do so. Expanding $(p - 1)(q - 1)$ algebraically we obtain $pq - p - q + 1 = n - p - q + 1$. Thus we know the value of $n - p - q + 1$, and so we can easily calculate the value of $p + q$ (since we know n). But we also know the value of pq, namely n. This gives us two simultaneous equations in two unknowns, and we can solve them using the quadratic formula. Here is an example. Suppose that we want to factor $n = 341$, and we are told that $(p - 1)(q - 1) = 300$. We want to find p and q. Following the argument just outlined, we know that $p + q = 341 + 1 - 300 = 42$. Plugging $q = 42 - p$ into $pq = 341$ we obtain $p(42 - p) = 341$, or $p^2 - 42p + 341 = 0$. The quadratic formula then tells us that $p = (42 + \sqrt{42^2 - 4 \cdot 341})/2 = 31$, and so the factors are 31 and $42 - 31 = 11$. Note that absolutely no trial divisions were involved here—it was just straight calculation.

25. First we translate UPLOAD into numbers: 2015 1114 0003. For each of these numbers, which we might call M, we need to compute $C = M^e$ **mod** $n = M^{17}$ **mod** 3233. Note that $n = 53 \cdot 61 = 3233$ and that $\gcd(e, (p-1)(q-1)) = \gcd(17, 52 \cdot 60) = 1$, as it should be. A computational aid tells us that 2015^{17} **mod** $3233 = 2545$, 1114^{17} **mod** $3233 = 2757$, and 0003^{17} **mod** $3233 = 1211$. Therefore the encrypted message is 2545 2757 1211.

27. This problem requires a great amount of calculation. Ideally, one should do it using a computer algebra package, such as *Mathematica* or *Maple*. Let us follow the procedure outlined in Example 9. It was computed there that the inverse of $e = 13$ modulo $n = 43 \cdot 59$ is $d = 937$. We need to compute 0667^{937} **mod** $2537 = 1808$, 1947^{937} **mod** $2537 = 1121$, and 0671^{937} **mod** $2537 = 0417$. (These calculations can in principle be done with a calculator, using the fast modular exponentiation algorithm, but it would probably take the better part of an hour and be prone to transcription errors.) Thus the original message is 1808 1121 0417, which is easily translated into letters as SILVER.

29. We follow the steps given in the text, with $p = 23$, $a = 5$, $k_1 = 8$, and $k_2 = 5$. Using *Maple*, we verify that 5 is a primitive root modulo 23, by noticing that 5^k as k runs from 0 to 21 produce distinct values (and of course 5^{22} **mod** $23 = 1$). We find that 5^8 **mod** $23 = 16$. So in Step (2), Alice sends 16 to Bob. Similarly, in Step (3), Bob sends 5^5 **mod** $23 = 20$ to Alice. In Step (4) Alice computes 20^8 **mod** $23 = 6$, and in Step (5) Bob computes 16^5 **mod** $23 = 6$. These are the same, of course, and thus 6 is the shared key.

31. See Example 10 for the procedure. First Alice translates her message into numbers: 1804 1111 0421 0417 2419 0708 1306. She then applies her decryption transformation sending each block x to x^{1183} **mod** 2867. (We should verify with *Maple* that $7 \cdot 1183$ **mod** $(60 \cdot 46) = 1$.) Using *Maple*, we see that the blocks become 1804^{1183} **mod** $2867 = 2186$, 1111^{1183} **mod** $2867 = 2087$, 0421^{1183} **mod** $2867 = 1279$, 0417^{1183} **mod** $2867 = 1251$, 2419^{1183} **mod** $2867 = 0326$, 0708^{1183} **mod** $2867 = 0816$, and 1306^{1183} **mod** $2867 = 1948$. If her friends apply Alice's encryption transformation to 2186 2087 1279 1251 0326 0816 1948, they will obtain the numbers of her original message.

33. Cathy knows the shared key $k_{\text{Alice,Bob}}$, but because she transmitted it to Alice encrypted, no one else knows it at the time Alice receives it. Alice can decrypt the first part of Cathy's message to find out what the key is. When Alice sends the second part of Cathy's message, which consists of $k_{\text{Alice,Bob}}$ encrypted with Bob's key, on to Bob, Bob can decrypt it to find the shared key, but it remains hidden from everyone else.

GUIDE TO REVIEW QUESTIONS FOR CHAPTER 4

1. Dividing 210 by 17 gives a quotient of 12 and a remainder of 6, which are the respective requested values.

2. **a)** $7 \mid a - b$ **b)** $0 \equiv -7$; $-1 \equiv -8$; $3 \equiv 17 \equiv -11$
c) $(10a + 13) - (-4b + 20) = 3(a - b) + 7(a + b - 1)$; note that 7 divides both terms

3. See Theorem 5 in Section 4.1.

4. See Example 5 in Section 4.2.

5. Octal: 154533; hexadecimal: D95B

6. 1110 1000 0110 and 1010 0000 1110 1011

7. See p. 258.

8. **a)** See Example 4 in Section 4.3 and the preceding paragraph on p. 258. **b)** $11^2 \cdot 23 \cdot 29$

CHAPTER 9
Relations

SECTION 9.1 Relations and Their Properties

This chapter is one of the most important in the book. Many structures in mathematics and computer science are formulated in terms of relations. Not only is the terminology worth learning, but the experience to be gained by working with various relations will prepare the student for the more advanced structures that he or she is sure to encounter in future work.

This section gives the basic terminology, especially the important notions of reflexivity, symmetry, antisymmetry, and transitivity. If we are given a relation as a set of ordered pairs, then reflexivity is easy to check for: we make sure that each element is related to itself. Symmetry is also fairly easy to test for: we make sure that no pair (a, b) is in the relation without its opposite (b, a) being present as well. To check for antisymmetry we make sure that no pair (a, b) with $a \neq b$ and its opposite are both in the relation. In other words, at most one of (a, b) and (b, a) is in the relation if $a \neq b$. Transitivity is much harder to verify, since there are many triples of elements to check. A common mistake to try to avoid is forgetting that a transitive relation that has pairs (a, b) and (b, a) must also include (a, a) and (b, b).

More importantly, we can be given a relation as a rule as to when elements are related. Exercises 4–7 are particularly useful in helping to understand the notions of reflexivity, symmetry, antisymmetry, and transitivity for relations given in this manner. Here you have to ask yourself the appropriate questions in order to determine whether the properties hold. Is every element related to itself? If so, the relation is reflexive. Are the roles of the variables in the definition interchangeable? If so, then the relation is symmetric. Does the definition preclude two different elements from each being related to the other? If so, then the relation is antisymmetric. Does the fact that one element is related to a second, which is in turn related to a third, mean that the first is related to the third? If so, then the relation is transitive.

In general, try to think of a relation in these two ways at the same time: as a set of ordered pairs and as a propositional function describing a relationship among objects.

1. In each case, we need to find all the pairs (a, b) with $a \in A$ and $b \in B$ such that the condition is satisfied. This is straightforward.

a) $\{(0,0), (1,1), (2,2), (3,3)\}$ **b)** $\{(1,3), (2,2), (3,1), (4,0)\}$

c) $\{(1,0), (2,0), (2,1), (3,0), (3,1), (3,2), (4,0), (4,1), (4,2), (4,3)\}$

d) Recall that $a \mid b$ means that b is a multiple of a (a is not allowed to be 0). Thus the answer is $\{(1,0), (1,1), (1,2), (1,3), (2,0), (2,2), (3,0), (3,3), (4,0)\}$.

e) We need to look for pairs whose greatest common divisor is 1—in other words, pairs that are relatively prime. Thus the answer is $\{(0,1), (1,0), (1,1), (1,2), (1,3), (2,1), (2,3), (3,1), (3,2), (4,1), (4,3)\}$.

f) There are not very many pairs of numbers (by definition only positive integers are considered) whose least common multiple is 2: only 1 and 2, and 2 and 2. Thus the answer is $\{(1,2), (2,1), (2,2)\}$.

3. a) This relation is not reflexive, since it does not include, for instance $(1,1)$. It is not symmetric, since it includes, for instance, $(2,4)$ but not $(4,2)$. It is not antisymmetric since it includes both $(2,3)$ and $(3,2)$, but $2 \neq 3$. It is transitive. To see this we have to check that *whenever* it includes (a,b) and (b,c), then it

also includes (a, c). We can ignore the element 1 since it never appears. If (a, b) is in this relation, then by inspection we see that a must be either 2 or 3. But $(2, c)$ and $(3, c)$ are in the relation for all $c \neq 1$; thus (a, c) has to be in this relation whenever (a, b) and (b, c) are. This proves that the relation is transitive. Note that it is very tedious to prove transitivity for an arbitrary list of ordered pairs.

b) This relation is reflexive, since all the pairs $(1, 1)$, $(2, 2)$, $(3, 3)$, and $(4, 4)$ are in it. It is clearly symmetric, the only nontrivial case to note being that both $(1, 2)$ and $(2, 1)$ are in the relation. It is not antisymmetric because both $(1, 2)$ and $(2, 1)$ are in the relation. It is transitive; the only nontrivial cases to note are that since both $(1, 2)$ and $(2, 1)$ are in the relation, we need to have (and do have) both $(1, 1)$ and $(2, 2)$ included as well.

c) This relation clearly is not reflexive and clearly is symmetric. It is not antisymmetric since both $(2, 4)$ and $(4, 2)$ are in the relation. It is not transitive, since although $(2, 4)$ and $(4, 2)$ are in the relation, $(2, 2)$ is not.

d) This relation is clearly not reflexive. It is not symmetric, since, for instance, $(1, 2)$ is included but $(2, 1)$ is not. It is antisymmetric, since there are no cases of (a, b) and (b, a) both being in the relation. It is not transitive, since although $(1, 2)$ and $(2, 3)$ are in the relation, $(1, 3)$ is not.

e) This relation is clearly reflexive and symmetric. It is trivially antisymmetric since there are no pairs (a, b) in the relation with $a \neq b$. It is trivially transitive, since the only time the hypothesis $(a, b) \in R \wedge (b, c) \in R$ is met is when $a = b = c$.

f) This relation is clearly not reflexive. The presence of $(1, 4)$ and absence of $(4, 1)$ shows that it is not symmetric. The presence of both $(1, 3)$ and $(3, 1)$ shows that it is not antisymmetric. It is not transitive; both $(2, 3)$ and $(3, 1)$ are in the relation, but $(2, 1)$ is not, for instance.

5. Recall the definitions: R is reflexive if $(a, a) \in R$ for all a; R is symmetric if $(a, b) \in R$ always implies $(b, a) \in R$; R is antisymmetric if $(a, b) \in R$ and $(b, a) \in R$ always implies $a = b$; and R is transitive if $(a, b) \in R$ and $(b, c) \in R$ always implies $(a, c) \in R$.

a) It is tautological that everyone who has visited Web page a has also visited Web page a, so R is reflexive. It is not symmetric, because there surely are Web pages a and b such that the set of people who visited a is a proper subset of the set of people who visited b (for example, the only link to page a may be on page b). Whether R is antisymmetric in truth is hard to say, but it is certainly conceivable that there are two different Web pages a and b that have had exactly the same set of visitors. In this case, $(a, b) \in R$ and $(b, a) \in R$, so R is not antisymmetric. Finally, R is transitive: if everyone who has visited a has also visited b, and everyone who has visited b has also visited c, then clearly everyone who has visited a has also visited c.

b) This relation is not reflexive, because for any page a that has links on it, $(a, a) \notin R$. The definition of R is symmetric in its very statement, so R is clearly symmetric. Also R is certainly not antisymmetric, because there surely are two different Web pages a and b out there that have no common links found on them. Finally, R is not transitive, because the two Web pages just mentioned, assuming they have links at all, give an example of the failure of the definition: $(a, b) \in R$ and $(b, a) \in R$, but $(a, a) \notin R$.

c) This relation is not reflexive, because for any page a that has no links on it, $(a, a) \notin R$. The definition of R is symmetric in its very statement, so R is clearly symmetric. Also R is certainly not antisymmetric, because there surely are two different Web pages a and b out there that have a common link found on them. Finally, R is surely not transitive. Page a might have only one link (say to this textbook), page c might have only one link different from this (say to the Erdős Number Project), and page b may have only the two links mentioned in this sentence. Then $(a, b) \in R$ and $(b, c) \in R$, but $(a, c) \notin R$.

d) This relation is probably not reflexive, because there probably exist Web pages out there with no links at all to them (for example, when they are in the process of being written and tested); for any such page a we have $(a, a) \notin R$. The definition of R is symmetric in its very statement, so R is clearly symmetric. Also R is certainly not antisymmetric, because there surely are two different Web pages a and b out there that are referenced by some third page. Finally, R is surely not transitive. Page a might have only one page that links

to it, page c might also have only one page, different from this, that links to it, and page b may be cited on both of these two pages. Then there would be no page that includes links to both pages a and c, so we have $(a,b) \in R$ and $(b,c) \in R$, but $(a,c) \notin R$.

7. **a)** This relation is not reflexive since it is not the case that $1 \neq 1$, for instance. It is symmetric: if $x \neq y$, then of course $y \neq x$. It is not antisymmetric, since, for instance, $1 \neq 2$ and also $2 \neq 1$. It is not transitive, since $1 \neq 2$ and $2 \neq 1$, for instance, but it is not the case that $1 \neq 1$.

b) This relation is not reflexive, since $(0,0)$ is not included. It is symmetric, because the commutative property of multiplication tells us that $xy = yx$, so that one of these quantities is greater than or equal to 1 if and only if the other is. It is not antisymmetric, since, for instance, $(2,3)$ and $(3,2)$ are both included. It is transitive. To see this, note that the relation holds between x and y if and only if either x and y are both positive or x and y are both negative. So assume that (a,b) and (b,c) are both in the relation. There are two cases, nearly identical. If a is positive, then so is b, since $(a,b) \in R$; therefore so is c, since $(b,c) \in R$, and hence $(a,c) \in R$. If a is negative, then so is b, since $(a,b) \in R$; therefore so is c, since $(b,c) \in R$, and hence $(a,c) \in R$.

c) This relation is not reflexive, since $(1,1)$ is not included, for instance. It is symmetric; the equation $x = y - 1$ is equivalent to the equation $y = x + 1$, which is the same as the equation $x = y + 1$ with the roles of x and y reversed. (A more formal proof of symmetry would be by cases. If x is related to y then either $x = y + 1$ or $x = y - 1$. In the former case, $y = x - 1$, so y is related to x; in the latter case $y = x + 1$, so y is related to x.) It is not antisymmetric, since, for instance, both $(1,2)$ and $(2,1)$ are in the relation. It is not transitive, since, for instance, although both $(1,2)$ and $(2,1)$ are in the relation, $(1,1)$ is not.

d) Recall that $x \equiv y \pmod 7$ means that $x - y$ is a multiple of 7, i.e., that $x - y = 7t$ for some integer t. This relation is reflexive, since $x - x = 7 \cdot 0$ for all x. It is symmetric, since if $x \equiv y \pmod 7$, then $x - y = 7t$ for some t; therefore $y - x = 7(-t)$, so $y \equiv x \pmod 7$. It is not antisymmetric, since, for instance, we have both $2 \equiv 9$ and $9 \equiv 2 \pmod 7$. It is transitive. Suppose $x \equiv y$ and $y \equiv z \pmod 7$. This means that $x - y = 7s$ and $y - z = 7t$ for some integers s and t. The trick is to add these two equations and note that the y disappears; we get $x - z = 7s + 7t = 7(s + t)$. By definition, this means that $x \equiv z \pmod 7$, as desired.

e) Every number is a multiple of itself (namely 1 times itself), so this relation is reflexive. (There is one bit of controversy here; we assume that 0 is to be considered a multiple of 0, even though we do not consider that 0 is a divisor of 0.) It is clearly not symmetric, since, for instance, 6 is a multiple of 2, but 2 is not a multiple of 6. The relation is not antisymmetric either; we have that 2 is a multiple of -2, for instance, and -2 is a multiple of 2, but $2 \neq -2$. The relation is transitive, however. If x is a multiple of y (say $x = ty$), and y is a multiple of z (say $y = sz$), then we have $x = t(sz) = (ts)z$, so we know that x is a multiple of z.

f) This relation is reflexive, since a and a are either both negative or both nonnegative. It is clearly symmetric from its form. It is not antisymmetric, since 5 is related to 6 and 6 is related to 5, but $5 \neq 6$. Finally, it is transitive, since if a is related to b and b is related to c, then all three of them must be negative, or all three must be nonnegative.

g) This relation is not reflexive, since, for instance, $17 \neq 17^2$. It is not symmetric, since although $289 = 17^2$, it is not the case that $17 = 289^2$. To see whether it is antisymmetric, suppose that we have both (x,y) and (y,x) in the relation. Then $x = y^2$ and $y = x^2$. To solve this system of equations, plug the second into the first, to obtain $x = x^4$, which is equivalent to $x - x^4 = 0$. The left-hand side factors as $x(1-x^3) = x(1-x)(1+x+x^2)$, so the solutions for x are 0 and 1 (and a pair of irrelevant complex numbers). The corresponding solutions for y are therefore also 0 and 1. Thus the only time we have both $x = y^2$ and $y = x^2$ is when $x = y$; this means that the relation is antisymmetric. It is not transitive, since, for example, $16 = 4^2$ and $4 = 2^2$, but $16 \neq 2^2$.

h) This relation is not reflexive, since, for instance, $17 \not\geq 17^2$. It is not symmetric, since although $289 \geq 17^2$, it is not the case that $17 \geq 289^2$. To see whether it is antisymmetric, we assume that both (x,y) and (y,x)

are in the relation. Then $x \geq y^2$ and $y \geq x^2$. Since both sides of the second inequality are nonnegative, we can square both sides to get $y^2 \geq x^4$. Combining this with the first inequality, we have $x \geq x^4$, which is equivalent to $x - x^4 \geq 0$. The left-hand side factors as $x(1 - x^3) = x(1 - x)(1 + x + x^2)$. The last factor is always positive, so we can divide the original inequality by it to obtain the equivalent inequality $x(1-x) \geq 0$. Now if $x > 1$ or $x < 0$, then the factors have different signs, so the inequality does not hold. Thus the only solutions are $x = 0$ and $x = 1$. The corresponding solutions for y are therefore also 0 and 1. Thus the only time we have both $x \geq y^2$ and $y \geq x^2$ is when $x = y$; this means that the relation is antisymmetric. It is transitive. Suppose $x \geq y^2$ and $y \geq z^2$. Again the second inequality implies that both sides are nonnegative, so we can square both sides to obtain $y^2 \geq z^4$. Combining these inequalities gives $x \geq z^4$. Now we claim that it is always the case that $z^4 \geq z^2$; if so, then we combine this fact with the last inequality to obtain $x \geq z^2$, so x is related to z. To verify the claim, note that since we are working with integers, it is always the case that $z^2 \geq |z|$ (equality for $z = 0$ and $z = 1$, strict inequality for other z). Squaring both sides gives the desired inequality.

9. Each of the properties is a universally quantified statement. Because the domain is empty, each of them is vacuously true.

11. The relations in parts **(a)**, **(b)**, and **(e)** all have at least one pair of the form (x, x) in them, so they are not irreflexive. The relations in parts **(c)**, **(d)**, and **(f)** do not, so they are irreflexive.

13. According to the preamble to Exercise 11, an irreflexive relation is one for which a is never related to itself; i.e., $\forall a((a, a) \notin R)$.

a) Since we saw in Exercise 5a that $\forall a((a, a) \in R)$, clearly R is not irreflexive.

b) Since there are probably pages a with no links at all, and for such pages it is true that there are no common links found on both page a and page a, this relation is probably not irreflexive.

c) This relation is not irreflexive, because for any page a that has links on it, $(a, a) \in R$.

d) This relation is not irreflexive, because for any page a that has links on it that are ever cited, $(a, a) \in R$.

15. The relation in Exercise 3a is neither reflexive nor irreflexive. It contains some of the pairs (a, a) but not all of them.

17. Of course many answers are possible. The empty relation is always irreflexive (x is never related to y). A less trivial example would be $(a, b) \in R$ if and only if a is taller than b. Since nobody is taller than him/herself, we always have $(a, a) \notin R$.

19. The relation in part **(a)** is asymmetric, since if a is taller than b, then certainly b cannot be taller than a. The relation in part **(b)** is not asymmetric, since there are many instances of a and b born on the same day (both cases in which $a = b$ and cases in which $a \neq b$), and in all such cases, it is also the case that b and a were born on the same day. The relations in part **(c)** and part **(d)** are just like that in part **(b)**, so they, too, are not asymmetric.

21. According to the preamble to Exercise 18, an asymmetric relation is one for which $(a, b) \in R$ and $(b, a) \in R$ can never hold simultaneously, even if $a = b$. Thus R is asymmetric if and only if R is antisymmetric and also irreflexive.

a) not asymmetric since $(-1, 1) \in R$ and $(1, -1) \in R$

b) not asymmetric since $(-1, 1) \in R$ and $(1, -1) \in R$

c) not asymmetric since $(-1, 1) \in R$ and $(1, -1) \in R$

d) not asymmetric since $(0, 0) \in R$

e) not asymmetric since $(2,1) \in R$ and $(1,2) \in R$

f) not asymmetric since $(0,1) \in R$ and $(1,0) \in R$

g) not asymmetric since $(1,1) \in R$

h) not asymmetric since $(2,1) \in R$ and $(1,2) \in R$

23. According to the preamble to Exercise 18, an asymmetric relation is one for which $(a,b) \in R$ and $(b,a) \in R$ can never hold simultaneously. In symbols, this is simply $\forall a \forall b \neg((a,b) \in R \wedge (b,a) \in R)$. Alternatively, $\forall a \forall b\,((a,b) \in R \rightarrow (b,a) \notin R)$.

25. There are mn elements of the set $A \times B$, if A is a set with m elements and B is a set with n elements. A relation from A to B is a subset of $A \times B$. Thus the question asks for the number of subsets of the set $A \times B$, which has mn elements. By the product rule, it is 2^{mn}.

27. **a)** By definition the answer is $\{\,(b,a) \mid a \text{ divides } b\,\}$, which, by changing the names of the dummy variables, can also be written $\{\,(a,b) \mid b \text{ divides } a\,\}$. (The universal set is still the set of positive integers.)

b) By definition the answer is $\{\,(a,b) \mid a \text{ does not divide } b\,\}$. (The universal set is still the set of positive integers.)

29. The inverse relation is just the graph of the inverse function. Somewhat more formally, we have $R^{-1} = \{\,(f(a),a) \mid a \in A\,\} = \{\,(b, f^{-1}(b)) \mid b \in B\,\}$, since we can index this collection just as easily by elements of B as by elements of A (using the correspondence $b = f(a)$).

31. This exercise is just a matter of the definitions of the set operations.

a) the set of pairs (a,b) where a is required to read b in a course or has read b

b) the set of pairs (a,b) where a is required to read b in a course and has read b

c) the set of pairs (a,b) where a is required to read b in a course or has read b, but not both; equivalently, the set of pairs (a,b) where a is required to read b in a course but has not done so, or has read b although not required to do so in a course

d) the set of pairs (a,b) where a is required to read b in a course but has not done so

e) the set of pairs (a,b) where a has read b although not required to do so in a course

33. To find $S \circ R$ we want to find the set of pairs (a,c) such that for some person b, a is a parent of b, and b is a sibling of c. Since brothers and sisters have the same parents, this means that a is also the parent of c. Thus $S \circ R$ is contained in the relation R. More specifically, $(a,c) \in S \circ R$ if and only if a is the parent of c, and c has a sibling (who is necessarily also a child of a). To find $R \circ S$ we want to find the set of pairs (a,c) such that for some person b, a is a sibling of b, and b is a parent of c. This is the same as the condition that a is the aunt or uncle of c (by blood, not marriage).

35. **a)** The union of two relations is the union of these sets. Thus $R_2 \cup R_4$ holds between two real numbers if R_2 holds or R_4 holds (or both, it goes without saying). Since it is always true that $a \leq b$ or $b \leq a$, $R_2 \cup R_4$ is all of $\mathbf{R}^2$, i.e., the relation that always holds.

b) For (a,b) to be in $R_3 \cup R_6$, we must have $a < b$ or $a \neq b$. Since this happens precisely when $a \neq b$, we see that the answer is R_6.

c) The intersection of two relations is the intersection of these sets. Thus $R_3 \cap R_6$ holds between two real numbers if R_3 holds and R_6 holds as well. Thus for (a,b) to be in $R_3 \cap R_6$, we must have $a < b$ and $a \neq b$. Since this happens precisely when $a < b$, we see that the answer is R_3.

d) For (a,b) to be in $R_4 \cap R_6$, we must have $a \leq b$ and $a \neq b$. Since this happens precisely when $a < b$, we see that the answer is R_3.

e) Recall that $R_3 - R_6 = R_3 \cap \overline{R_6}$. But $\overline{R_6} = R_5$, so we are asked for $R_3 \cap R_5$. It is impossible for $a < b$ and $a = b$ to hold at the same time, so the answer is $\emptyset$, i.e., the relation that never holds.

f) Reasoning as in part **(e)**, we want $R_6 \cap \overline{R_3} = R_6 \cap R_2$, which is clearly R_1 (since $a \neq b$ and $a \geq b$ precisely when $a > b$).

g) Recall that $R_2 \oplus R_6 = (R_2 \cap \overline{R_6}) \cup (R_6 \cap \overline{R_2})$. We see that $R_2 \cap \overline{R_6} = R_2 \cap R_5 = R_5$, and $R_6 \cap \overline{R_2} = R_6 \cap R_3 = R_3$. Thus our answer is $R_5 \cup R_3 = R_4$.

h) Recall that $R_3 \oplus R_5 = (R_3 \cap \overline{R_5}) \cup (R_5 \cap \overline{R_3})$. We see that $R_3 \cap \overline{R_5} = R_3 \cap R_6 = R_3$, and $R_5 \cap \overline{R_3} = R_5 \cap R_2 = R_5$. Thus our answer is $R_3 \cup R_5 = R_4$.

37. Recall that the composition of two relations all defined on a common set is defined as follows: $(a, c) \in S \circ R$ if and only if there is some element b such that $(a, b) \in R$ and $(b, c) \in S$. We have to apply this in each case.

a) For (a, c) to be in $R_2 \circ R_1$, we must find an element b such that $(a, b) \in R_1$ and $(b, c) \in R_2$. This means that $a > b$ and $b \geq c$. Clearly this can be done if and only if $a > c$ to begin with. But that is precisely the statement that $(a, c) \in R_1$. Therefore we have $R_2 \circ R_1 = R_1$.

b) For (a, c) to be in $R_2 \circ R_2$, we must find an element b such that $(a, b) \in R_2$ and $(b, c) \in R_2$. This means that $a \geq b$ and $b \geq c$. Clearly this can be done if and only if $a \geq c$ to begin with. But that is precisely the statement that $(a, c) \in R_2$. Therefore we have $R_2 \circ R_2 = R_2$. In particular, this shows that R_2 is transitive.

c) For (a, c) to be in $R_3 \circ R_5$, we must find an element b such that $(a, b) \in R_5$ and $(b, c) \in R_3$. This means that $a = b$ and $b < c$. Clearly this can be done if and only if $a < c$ to begin with (choose $b = a$). But that is precisely the statement that $(a, c) \in R_3$. Therefore we have $R_3 \circ R_5 = R_3$. One way to look at this is to say that R_5, the equality relation, acts as an identity for the composition operation (on the right—although it is also an identity on the left as well).

d) For (a, c) to be in $R_4 \circ R_1$, we must find an element b such that $(a, b) \in R_1$ and $(b, c) \in R_4$. This means that $a > b$ and $b \leq c$. Clearly this can always be done simply by choosing b to be small enough. Therefore we have $R_4 \circ R_1 = \mathbf{R}^2$, the relation that always holds.

e) For (a, c) to be in $R_5 \circ R_3$, we must find an element b such that $(a, b) \in R_3$ and $(b, c) \in R_5$. This means that $a < b$ and $b = c$. Clearly this can be done if and only if $a < c$ to begin with (choose $b = c$). But that is precisely the statement that $(a, c) \in R_3$. Therefore we have $R_5 \circ R_3 = R_3$. One way to look at this is to say that R_5, the equality relation, acts as an identity for the composition operation (on the left—although it is also an identity on the right as well).

f) For (a, c) to be in $R_3 \circ R_6$, we must find an element b such that $(a, b) \in R_6$ and $(b, c) \in R_3$. This means that $a \neq b$ and $b < c$. Clearly this can always be done simply by choosing b to be small enough. Therefore we have $R_3 \circ R_6 = \mathbf{R}^2$, the relation that always holds.

g) For (a, c) to be in $R_4 \circ R_6$, we must find an element b such that $(a, b) \in R_6$ and $(b, c) \in R_4$. This means that $a \neq b$ and $b \leq c$. Clearly this can always be done simply by choosing b to be small enough. Therefore we have $R_4 \circ R_6 = \mathbf{R}^2$, the relation that always holds.

h) For (a, c) to be in $R_6 \circ R_6$, we must find an element b such that $(a, b) \in R_6$ and $(b, c) \in R_6$. This means that $a \neq b$ and $b \neq c$. Clearly this can always be done simply by choosing b to be something other than a or c. Therefore we have $R_6 \circ R_6 = \mathbf{R}^2$, the relation that always holds. Note that since the answer is not R_6 itself, we know that R_6 is not transitive.

39. One earns a doctorate by, among other things, writing a thesis under an advisor, so this relation makes sense. (We ignore anomalies like someone having two advisors or someone being awarded a doctorate without having an advisor.) For (a, b) to be in R^2, we must find a c such that $(a, c) \in R$ and $(c, b) \in R$. In our context, this says that b got his/her doctorate under someone who got his/her doctorate under a. Colloquially, a is the academic grandparent of b, or b is the academic grandchild of a. Generalizing, $(a, b) \in R^n$ precisely when there is a sequence of $n+1$ people, starting with a and ending with b, such that each is the advisor of the next person

in the sequence. People with doctorates like to look at these sequences (and trace their ancestry) back as far as they can. There is an excellent website for doing so in mathematics (`www.genealogy.math.ndsu.nodak.edu`).

41. **a)** The union of two relations is the union of these sets. Thus $R_1 \cup R_2$ holds between two integers if R_1 holds or R_2 holds (or both, it goes without saying). Thus $(a, b) \in R_1 \cup R_2$ if and only if $a \equiv b \pmod 3$ or $a \equiv b \pmod 4$. There is not a good easier way to state this, other than perhaps to say that $a - b$ is a multiple of either 3 or 4, or to work modulo 12 and write $a - b \equiv 0, 3, 4, 6, 8,$ or $9 \pmod{12}$.

b) The intersection of two relations is the intersection of these sets. Thus $R_1 \cap R_2$ holds between two integers if R_1 holds and R_2 holds. Thus $(a, b) \in R_1 \cap R_2$ if and only if $a \equiv b \pmod 3$ and $a \equiv b \pmod 4$. Since this means that $a - b$ is a multiple of both 3 and 4, and that happens if and only if $a - b$ is a multiple of 12, we can state this more simply as $a \equiv b \pmod{12}$.

c) By definition $R_1 - R_2 = R_1 \cap \overline{R_2}$. Thus this relation holds between two integers if R_1 holds and R_2 does not hold. We can write this in symbols by saying that $(a, b) \in R_1 - R_2$ if and only if $a \equiv b \pmod 3$ and $a \not\equiv b \pmod 4$. We could, if we wished, state this working modulo 12: $(a, b) \in R_1 - R_2$ if and only if $a - b \equiv 3, 6,$ or $9 \pmod{12}$.

d) By definition $R_2 - R_1 = R_2 \cap \overline{R_1}$. Thus this relation holds between two integers if R_2 holds and R_1 does not hold. We can write this in symbols by saying that $(a, b) \in R_2 - R_1$ if and only if $a \equiv b \pmod 4$ and $a \not\equiv b \pmod 3$. We could, if we wished, state this working modulo 12: $(a, b) \in R_2 - R_1$ if and only if $a - b \equiv 4$ or $8 \pmod{12}$.

e) We know that $R_1 \oplus R_2 = (R_1 - R_2) \cup (R_2 - R_1)$, so we look at our solutions to part **(c)** and part **(d)**. Thus this relation holds between two integers if R_1 holds and R_2 does not hold, or vice versa. We can write this in symbols by saying that $(a, b) \in R_1 \oplus R_2$ if and only if ($a \equiv b \pmod 3$ and $a \not\equiv b \pmod 4$) or ($a \equiv b \pmod 4$ and $a \not\equiv b \pmod 3$). We could, if we wished, state this working modulo 12: $(a, b) \in R_1 \oplus R_2$ if and only if $a - b \equiv 3, 4, 6, 8$ or $9 \pmod{12}$. We could also say that $a - b$ is a multiple of 3 or 4 but not both.

43. A relation is just a subset. A subset can either contain a specified element or not; half of them do and half of them do not. Therefore 8 of the 16 relations on $\{0, 1\}$ contain the pair $(0, 1)$. Alternatively, a relation on $\{0, 1\}$ containing the pair $(0, 1)$ is just a set of the form $\{(0, 1)\} \cup X$, where $X \subseteq \{(0, 0), (1, 0), (1, 1)\}$. Since this latter set has 3 elements, it has $2^3 = 8$ subsets.

45. This is similar to Example 16 in this section.

a) A relation on a set S with n elements is a subset of $S \times S$. Since $S \times S$ has n^2 elements, we are asking for the number of subsets of a set with n^2 elements, which is 2^{n^2}. In our case $n = 4$, so the answer is $2^{16} = 65{,}536$.

b) In solving part **(a)**, we had 16 binary choices to make—whether to include a pair (x, y) in the relation or not as x and y ranged over the set $\{a, b, c, d\}$. In this part, one of those choices has been made for us: we *must* include (a, a). We are free to make the other 15 choices. So the answer is $2^{15} = 32{,}768$. See Exercise 47 for more problems of this type.

47. These are combinatorics problems, some harder than others. Let A be the set with n elements on which the relations are defined.

a) To specify a symmetric relation, we need to decide, for each unordered pair $\{a, b\}$ of distinct elements of A, whether to include the pairs (a, b) and (b, a) or leave them out; this can be done in 2 ways for each such unordered pair. Also, for each element $a \in A$, we need to decide whether to include (a, a) or not, again 2 possibilities. We can think of these two parts as one by considering an element to be an unordered pair with repetition allowed. Thus we need to make this 2-fold choice $C(n + 1, 2)$ times, since there are $C(n + 2 - 1, 2)$ ways to choose an unordered pair with repetition allowed. Therefore the answer is $2^{C(n+1,2)} = 2^{n(n+1)/2}$.

b) This is somewhat similar to part **(a)**. For each unordered pair $\{a, b\}$ of distinct elements of A, we have a 3-way choice—either include (a, b) only, include (b, a) only, or include neither. For each element of A we have a 2-way choice. Therefore the answer is $3^{C(n,2)}2^n = 3^{n(n-1)/2}2^n$.

c) As in part **(b)** we have a 3-way choice for $a \neq b$. There is no choice about including (a, a) in the relation—the definition prohibits it. Therefore the answer is $3^{C(n,2)} = 3^{n(n-1)/2}$.

d) For each ordered pair (a, b), with $a \neq b$ (and there are $P(n, 2)$ such pairs), we can choose to include (a, b) or to leave it out. There is no choice for pairs (a, a). Therefore the answer is $2^{P(n,2)} = 2^{n(n-1)}$.

e) This is just like part **(a)**, except that there is no choice about including (a, a). For each unordered pair of distinct elements of A, we can choose to include neither or both of the corresponding ordered pairs. Therefore the answer is $2^{C(n,2)} = 2^{n(n-1)/2}$.

f) We have complete freedom with the ordered pairs (a, b) with $a \neq b$, so that part of the choice gives us $2^{P(n,2)}$ possibilities, just as in part **(d)**. For the decision as to whether to include (a, a), two of the 2^n possibilities are prohibited: we cannot include all such pairs, and we cannot leave them all out. Therefore the answer is $2^{P(n,2)}(2^n - 2) = 2^{n^2-n}(2^n - 2) = 2^{n^2} - 2^{n^2-n+1}$.

49. The second sentence of the proof asks us to "take an element $b \in A$ such that $(a, b) \in R$." There is no guarantee that such an element exists for the taking. This is the only mistake in the proof. If one could be guaranteed that each element in A is related to at least one element, then symmetry and transitivity would indeed imply reflexivity. Without this assumption, however, the proof and the proposition are wrong. As a simple example, take the relation $\emptyset$ on any nonempty set. This relation is vacuously symmetric and transitive, but not reflexive. Here is another counterexample: the relation $\{(1, 1), (1, 2), (2, 1), (2, 2)\}$ on the set $\{1, 2, 3\}$.

51. We need to show two things. First, we need to show that if a relation R is symmetric, then $R = R^{-1}$, which means we must show that $R \subseteq R^{-1}$ and $R^{-1} \subseteq R$. To do this, let $(a, b) \in R$. Since R is symmetric, this implies that $(b, a) \in R$. But since R^{-1} consists of all pairs (a, b) such that $(b, a) \in R$, this means that $(a, b) \in R^{-1}$. Thus we have shown that $R \subseteq R^{-1}$. Next let $(a, b) \in R^{-1}$. By definition this means that $(b, a) \in R$. Since R is symmetric, this implies that $(a, b) \in R$ as well. Thus we have shown that $R^{-1} \subseteq R$.

Second we need to show that $R = R^{-1}$ implies that R is symmetric. To this end we let $(a, b) \in R$ and try to show that (b, a) is also necessarily an element of R. Since $(a, b) \in R$, the definition tells us that $(b, a) \in R^{-1}$. But since we are under the hypothesis that $R = R^{-1}$, this tells us that $(b, a) \in R$, exactly as desired.

53. Suppose that R is reflexive. We must show that R^{-1} is reflexive, i.e., that $(a, a) \in R^{-1}$ for each $a \in A$. Now since R is reflexive, we know that $(a, a) \in R$ for each $a \in R$. By definition, this tells us that $(a, a) \in R^{-1}$, as desired. (Interchanging the two a's in the pair (a, a) leaves it as it was.) Conversely, if R^{-1} is reflexive, then $(a, a) \in R^{-1}$ for each $a \in A$. By definition this means that $(a, a) \in R$ (again we interchanged the two a's).

55. We prove this by induction on n. The case $n = 1$ is trivial, since it is the statement $R = R$. Assume the inductive hypothesis that $R^n = R$. We must show that $R^{n+1} = R$. By definition $R^{n+1} = R^n \circ R$. Thus our task is to show that $R^n \circ R \subseteq R$ and $R \subseteq R^n \circ R$. The first uses the transitivity of R, as follows. Suppose that $(a, c) \in R^n \circ R$. This means that there is an element b such that $(a, b) \in R$ and $(b, c) \in R^n$. By the inductive hypothesis, the latter statement implies that $(b, c) \in R$. Thus by the transitivity of R, we know that $(a, c) \in R$, as desired.

Next assume that $(a, b) \in R$. We must show that $(a, b) \in R^n \circ R$. By the inductive hypothesis, $R^n = R$, and therefore R^n is reflexive by assumption. Thus $(b, b) \in R^n$. Since we have $(a, b) \in R$ and $(b, b) \in R^n$, we have by definition that (a, b) is an element of $R^n \circ R$, exactly as desired. (The first half of this proof was not really necessary, since Theorem 1 in this section already told us that $R^n \subseteq R$ for all n.)

there (forming the first set of m-tuples). A simple example would be to let $R = \{(a, b)\}$ and $S = \{(a, c)\}$, $n = 2$, $m = 1$, and $i_1 = 1$. Then $R - S = R$, so $P_1(R - S) = P_1(R) = \{(a)\}$. On the other hand, $P_1(R) = P_1(S) = \{(a)\}$, so $P_1(R) - P_1(S) = \emptyset$.

29. This is similar to Example 13.

a) Since two databases are listed in the "FROM" field, the first operation is to form the join of these two databases, specifically the join J_2 of these two databases. We then apply the selection operator with the condition "Quantity ≤ 10." This join will have eight 5-tuples in it. Finally we want just the Supplier and Project, so we are forming the projection $P_{1,3}$.

b) Four of the 5-tuples in the joined database have a quantity of no more than 10. The output, then, is the set of the four 2-tuples corresponding to these fields: $(23, 1), (23, 3), (31, 3), (32, 4)$.

31. A primary key is a domain whose value determines the values of all the other domains. For this relation, this does not happen. The third domain (the modulus) is not a primary key, because, for example, $1 \equiv 11 \pmod{10}$ and $2 \equiv 12 \pmod{10}$, so the triples $(1, 11, 10)$ and $(2, 12, 10)$ are both in the relation. Knowing that the third component of a triple is 10 does not tell us what the other two components are. Similarly, the triples $(1, 11, 10)$ and $(1, 21, 10)$ are both in the relation, so the first domain is not a key; and the triples $(1, 11, 10)$ and $(11, 11, 10)$ are both in the relation, so the second domain is not a key.

SECTION 9.3 Representing Relations

Matrices and directed graphs provide useful ways for computers and humans to represent relations and manipulate them. Become familiar with working with these representations and the operations on them (especially the matrix operation for forming composition) by working these exercises. Some of these exercises explore how properties of a relation can be found from these representations.

1. In each case we use a 3×3 matrix, putting a 1 in position (i, j) if the pair (i, j) is in the relation and a 0 in position (i, j) if the pair (i, j) is not in the relation. For instance, in part **(a)** there are 1's in the first row, since each of the pairs $(1, 1)$, $(1, 2)$, and $(1, 3)$ are in the relation, and there are 0's elsewhere.

a) $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ **b)** $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ **c)** $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$ **d)** $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

3. a) Since the $(1, 1)^{\text{th}}$ entry is a 1, $(1, 1)$ is in the relation. Since $(1, 2)^{\text{th}}$ entry is a 0, $(1, 2)$ is not in the relation. Continuing in this manner, we see that the relation contains $(1, 1)$, $(1, 3)$, $(2, 2)$, $(3, 1)$, and $(3, 3)$.
b) $(1, 2)$, $(2, 2)$, and $(3, 2)$ **c)** $(1, 1)$, $(1, 2)$, $(1, 3)$, $(2, 1)$, $(2, 3)$, $(3, 1)$, $(3, 2)$, and $(3, 3)$

5. An irreflexive relation (see the preamble to Exercise 11 in Section 9.1) is one in which no element is related to itself. In the matrix, this means that there are no 1's on the main diagonal (position m_{ii} for some i). Equivalently, the relation is irreflexive if and only if every entry on the main diagonal of the matrix is 0.

7. For reflexivity we want all 1's on the main diagonal; for irreflexivity we want all 0's on the main diagonal; for symmetry, we want the matrix to be symmetric about the main diagonal (equivalently, the matrix equals its transpose); for antisymmetry we want there never to be two 1's symmetrically placed about the main diagonal (equivalently, the meet of the matrix and its transpose has no 1's off the main diagonal); and for transitivity we want the Boolean square of the matrix (the Boolean product of the matrix and itself) to be "less than or equal to" the original matrix in the sense that there is a 1 in the original matrix at every location where there is a 1 in the Boolean square.

a) Since there are all 1's on the main diagonal, this relation is reflexive and not irreflexive. Since the matrix is symmetric, the relation is symmetric. The relation is not antisymmetric—look at positions $(1,3)$ and $(3,1)$. Finally, the Boolean square of this matrix is itself, so the relation is transitive.

b) Since there are both 0's and 1's on the main diagonal, this relation is neither reflexive nor irreflexive. Since the matrix is not symmetric, the relation is not symmetric (look at positions $(1,2)$ and $(2,1)$, for example). The relation is antisymmetric since there are never two 1's symmetrically placed with respect to the main diagonal. Finally, the Boolean square of this matrix is itself, so the relation is transitive.

c) Since there are both 0's and 1's on the main diagonal, this relation is neither reflexive nor irreflexive. Since the matrix is symmetric, the relation is symmetric. The relation is not antisymmetric—look at positions $(1,3)$ and $(3,1)$, for example. Finally, the Boolean square of this matrix is the matrix with all 1's, so the relation is not transitive (1 is related to 2, and 2 is related to 1, but 2 is not related to 2).

9. Note that the total number of entries in the matrix is $100^2 = 10{,}000$.

a) There is a 1 in the matrix for each pair of distinct positive integers not exceeding 100, namely in position (a,b) where $a > b$. Thus the answer is the number of subsets of size 2 from a set of 100 elements, i.e., $C(100,2) = 4950$.

b) There is a 1 in the matrix at each position except the 100 positions on the main diagonal. Therefore the answer is $100^2 - 100 = 9900$.

c) There is a 1 in the matrix at each entry just below the main diagonal (i.e., in positions $(2,1)$, $(3,2)$, ..., $(100,99)$. Therefore the answer is 99.

d) The entire first row of this matrix corresponds to $a = 1$. Therefore the matrix has 100 nonzero entries.

e) This relation has only the one element $(1,1)$ in it, so the matrix has just one nonzero entry.

11. Since the relation $\overline{R}$ is the relation that contains the pair (a,b) (where a and b are elements of the appropriate sets) if and only if R does not contain that pair, we can form the matrix for $\overline{R}$ simply by changing all the 1's to 0's and 0's to 1's in the matrix for R.

13. Exercise 12 tells us how to do part **(a)** (we take the transpose of the given matrix $\mathbf{M}_R$, which in this case happens to be the matrix itself). Exercise 11 tells us how to do part **(b)** (we change 1's to 0's and 0's to 1's in $\mathbf{M}_R$). For part **(c)** we take the Boolean product of $\mathbf{M}_R$ with itself.

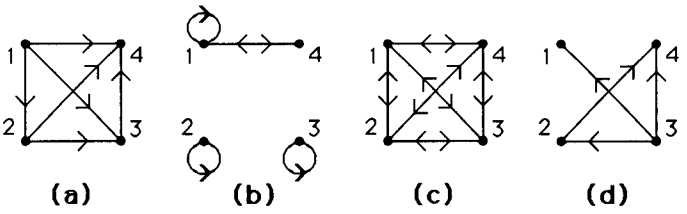
a) $\begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ **b)** $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ **c)** $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

15. We compute the Boolean powers of $\mathbf{M}_R$; thus $\mathbf{M}_{R^2} = \mathbf{M}_R^{[2]} = \mathbf{M}_R \odot \mathbf{M}_R$, $\mathbf{M}_{R^3} = \mathbf{M}_R^{[3]} = \mathbf{M}_R \odot \mathbf{M}_R^{[2]}$, and $\mathbf{M}_{R^4} = \mathbf{M}_R^{[4]} = \mathbf{M}_R \odot \mathbf{M}_R^{[3]}$.

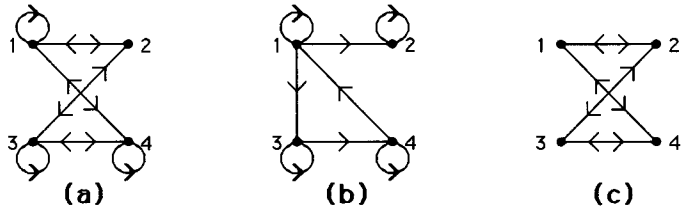
a) $\begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$ **b)** $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ **c)** $\begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

17. The matrix for the complement has a 1 wherever the matrix for the relation has a 0, and vice versa. Therefore the number of nonzero entries in $\mathbf{M}_{\overline{R}}$ is $n^2 - k$, since these matrices have n rows and n columns.

19. In each case we need a vertex for each of the elements, and we put in a directed edge from x to y if there is a 1 in position (x,y) of the matrix. For simplicity we have indicated pairs of edges between the same two vertices in opposite directions by using a double arrowhead, rather than drawing two separate lines.



21. In each case we need a vertex for each of the elements, and we put in a directed edge from x to y if there is a 1 in position (x, y) of the matrix. For simplicity we have indicated pairs of edges between the same two vertices in opposite directions by using a double arrowhead, rather than drawing two separate lines.



23. We list all the pairs (x, y) for which there is an edge from x to y in the directed graph: $\{(a, b), (a, c), (b, c), (c, b)\}$.

25. We list all the pairs (x, y) for which there is an edge from x to y in the directed graph: $\{(a, c), (b, a), (c, d), (d, b)\}$.

27. We list all the pairs (x, y) for which there is an edge from x to y in the directed graph: $\{(a, a), (a, b), (a, c), (b, a), (b, b), (b, c), (c, a), (c, b), (d, d)\}$.

29. An asymmetric relation is one for which it never happens that a is related to b and simultaneously b is related to a, even when $a = b$. In terms of the directed graph, this means that we must see no loops and no closed paths of length 2 (i.e., no pairs of edges between two vertices going in opposite directions).

31. Recall that the relation is reflexive if there is a loop at each vertex; irreflexive if there are no loops at all; symmetric if edges appear only in **antiparallel** pairs (edges from one vertex to a second vertex and from the second back to the first); antisymmetric if there is no pair of antiparallel edges; and transitive if all paths of length 2 (a pair of edges (x, y) and (y, z)) are accompanied by the corresponding path of length 1 (the edge (x, z)). The relation drawn in Exercise 23 is not reflexive but is irreflexive since there are no loops. It is not symmetric, since, for instance, the edge (a, b) is present but not the edge (b, a). It is not antisymmetric, since both edges (b, c) and (c, b) are present. It is not transitive, since the path $(b, c), (c, b)$ from b to b is not accompanied by the edge (b, b). The relation drawn in Exercise 24 is reflexive and not irreflexive since there is a loop at each vertex. It is not symmetric, since, for instance, the edge (b, a) is present but not the edge (a, b). It is antisymmetric, since there are no pairs of antiparallel edges. It is transitive, since the only nontrivial path of length 2 is bac, and the edge (b, c) is present. The relation drawn in Exercise 25 is not reflexive but is irreflexive since there are no loops. It is not symmetric, since, for instance, the edge (b, a) is present but not the edge (a, b). It is antisymmetric, since there are no pairs of antiparallel edges. It is not transitive, since the edges (a, c) and (c, d) are present, but not (a, d).

33. Since the inverse relation consists of all pairs (b, a) for which (a, b) is in the original relation, we just have to take the digraph for R and reverse the direction on every edge.

35. We prove this statement by induction on n. The basis step $n = 1$ is tautologically true, since $\mathbf{M}_R^{[1]} = \mathbf{M}_R$. Assume the inductive hypothesis that $\mathbf{M}_R^{[n]}$ is the matrix representing R^n. Now $\mathbf{M}_R^{[n+1]} = \mathbf{M}_R \odot \mathbf{M}_R^{[n]}$. By the inductive hypothesis and the assertion made before Example 5, that $\mathbf{M}_{S \circ R} = \mathbf{M}_R \odot \mathbf{M}_S$, the right-hand side is the matrix representing $R^n \circ R$. But $R^n \circ R = R^{n+1}$, so our proof is complete.

SECTION 9.4 Closures of Relations

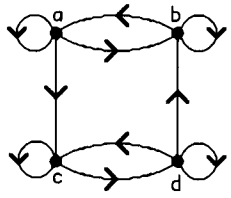
This section is harder than the previous ones in this chapter. Warshall's algorithm, in particular, is fairly tricky, and Exercise 27 should be worked carefully, following Example 8. It is easy to forget to include the loops (a, a) when forming transitive closures "by hand."

1. **a)** The reflexive closure of R is R together with all the pairs (a, a). Thus in this case the closure of R is $\{(0,0), (0,1), (1,1), (1,2), (2,0), (2,2), (3,0), (3,3)\}$.

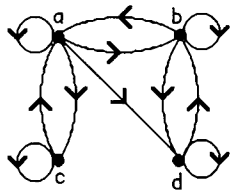
b) The symmetric closure of R is R together with all the pairs (b, a) for which (a, b) is in R. For example, since $(1,2)$ is in R, we need to add $(2,1)$. Thus the closure of R is $\{(0,1), (0,2), (0,3), (1,0), (1,1), (1,2), (2,0), (2,1), (2,2), (3,0)\}$.

3. To form the symmetric closure we need to add all the pairs (b, a) such that (a, b) is in R. In this case, that means that we need to include pairs (b, a) such that a divides b, which is equivalent to saying that we need to include all the pairs (a, b) such that b divides a. Thus the closure is $\{(a, b) \mid a \text{ divides } b \text{ or } b \text{ divides } a\}$.

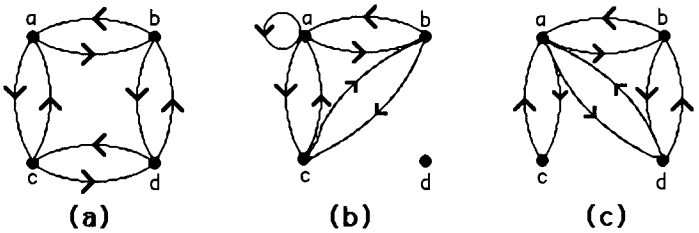
5. We form the reflexive closure by taking the given directed graph and appending loops at all vertices at which there are not already loops.



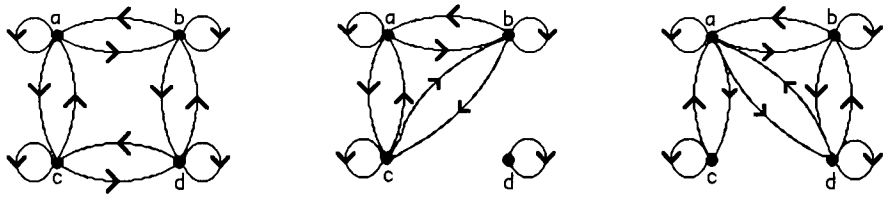
7. We form the reflexive closure by taking the given directed graph and appending loops at all vertices at which there are not already loops.



9. We form the symmetric closure by taking the given directed graph and appending an edge pointing in the opposite direction for every edge already in the directed graph (unless it is already there); in other words, we append the edge (b, a) whenever we see the edge (a, b). We have labeled the figures below (a), (b), and (c), corresponding to Exercises 5, 6, and 7, respectively.



11. We are asked for the symmetric and reflexive closure of the given relation. We form it by taking the given directed graph and appending (1) a loop at each vertex at which there is not already a loop and (2) an edge pointing in the opposite direction for every edge already in the directed graph (unless it is already there). We have labeled the figures below (a), (b), and (c), corresponding to Exercises 5, 6, and 7, respectively.



13. The symmetric closure of R is $R \cup R^{-1}$. The matrix for R^{-1} is $\mathbf{M}_R^t$, as we saw in Exercise 12 in Section 9.3. The matrix for the union of two relations is the join of the matrices for the two relations, as we saw in Section 9.3. Therefore the matrix representing the symmetric closure of R is indeed $\mathbf{M}_R \vee \mathbf{M}_R^t$.

15. If R is already irreflexive, then it is clearly its own irreflexive closure. On the other hand if R is not irreflexive, then there is no relation containing R that is irreflexive, since the loop or loops in R prevent any such relation from being irreflexive. Thus in this case R has no irreflexive closure. This exercise shows essentially that the concept of "irreflexive closure" is rather useless, since no relation has one unless it is already irreflexive (in which case it is its own "irreflexive closure").

17. A circuit of length 3 can be written as a sequence of 4 vertices, each joined to the next by an edge of the given directed graph, ending at the same vertex at which it began. There are several such circuits here, and we just have to be careful and systematically list them all. There are the circuits formed entirely by the loops: *aaaa*, *cccc*, and *eeee*. The triangles *abea* and *adea* also qualify. Two circuits start at *b*: *bccb* and *beab*. There are two more circuits starting at *c*, namely *ccbc* and *cbcc*. Similarly there are the circuits *deed*, *eede* and *edee*, as well as the other trips around the triangle: *eabe*, *dead*, and *eade*.

19. The way to form these powers is first to form the matrix representing R, namely

$$\mathbf{M}_R = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix},$$

and then take successive Boolean powers of it to get the matrices representing R^2, R^3, and so on. Finally, for part **(f)** we take the join of the matrices representing R, R^2, ..., R^5. Since the matrix is a perfectly good way to express the relation, we will not list the ordered pairs.

a) The matrix for R^2 is the Boolean product of the matrix displayed above with itself, namely

$$\mathbf{M}_{R^2} = \mathbf{M}_R^{[2]} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

b) The matrix for R^3 is the Boolean product of the first matrix displayed above with the answer to part **(a)**, namely

$$\mathbf{M}_{R^3} = \mathbf{M}_R^{[3]} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

c) The matrix for R^4 is the Boolean product of the first matrix displayed above with the answer to part **(b)**, namely

$$\mathbf{M}_{R^4} = \mathbf{M}_R^{[4]} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

d) The matrix for R^5 is the Boolean product of the first matrix displayed above with the answer to part **(c)**, namely

$$\mathbf{M}_{R^5} = \mathbf{M}_R^{[5]} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

e) The matrix for R^6 is the Boolean product of the first matrix displayed above with the answer to part **(d)**, namely

$$\mathbf{M}_{R^6} = \mathbf{M}_R^{[6]} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

f) The matrix for R^* is the join of the first matrix displayed above and the answers to parts **(a)** through **(d)**, namely

$$\mathbf{M}_{R^*} = \mathbf{M}_R \vee \mathbf{M}_R^{[2]} \vee \mathbf{M}_R^{[3]} \vee \mathbf{M}_R^{[4]} \vee \mathbf{M}_R^{[5]} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

21. a) The pair (a, b) is in R^2 if there is a person c other than a or b who is in a class with a and a class with b. Note that it is almost certain that (a, a) is in R^2, since as long as a is taking a class that has at least one other person in it, that person serves as the "c."

b) The pair (a, b) is in R^3 if there are persons c (different from a) and d (different from b and c) such that c is in a class with a, c is in a class with d, and d is in a class with b.

c) The pair (a, b) is in R^* if there is a sequence of persons, $c_0, c_1, c_2, \ldots, c_n$, with $n \geq 1$, such that $c_0 = a$, $c_n = b$, and for each i from 1 to n, $c_{i-1} \neq c_i$ and c_{i-1} is in at least one class with c_i.

23. Suppose that $(a, b) \in R^*$; then there is a path from a to b in (the digraph for) R. Given such a path, if R is symmetric, then the reverse of every edge in the path is also in R; therefore there is a path from b to a in R (following the given path backwards). This means that (b, a) is in R^* whenever (a, b) is, exactly what we needed to prove.

25. Algorithm 1 finds the transitive closure by computing the successive powers and taking their join. We exhibit our answers in matrix form as $\mathbf{M}_R \vee \mathbf{M}_R^{[2]} \vee \ldots \vee \mathbf{M}_R^{[n]} = \mathbf{M}_{R^*}$.

a)
$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \vee \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \vee \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \vee \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

b)
$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \vee \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \vee \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \vee \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

c) $\begin{bmatrix}0&1&1&1\\0&0&1&1\\0&0&0&1\\0&0&0&0\end{bmatrix} \vee \begin{bmatrix}0&0&1&1\\0&0&0&1\\0&0&0&0\\0&0&0&0\end{bmatrix} \vee \begin{bmatrix}0&0&0&1\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix} \vee \begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix} = \begin{bmatrix}0&1&1&1\\0&0&1&1\\0&0&0&1\\0&0&0&0\end{bmatrix}$

Note that the relation was already transitive, so its transitive closure is itself.

d) $\begin{bmatrix}1&0&0&1\\1&0&1&0\\1&1&0&1\\0&1&0&0\end{bmatrix} \vee \begin{bmatrix}1&1&0&1\\1&1&0&1\\1&1&1&1\\1&0&1&0\end{bmatrix} \vee \begin{bmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&0&1\end{bmatrix} \vee \begin{bmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix} = \begin{bmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}$

27. In Warshall's algorithm (Algorithm 2 in this section), we compute a sequence of matrices $\mathbf{W}_0$ (the matrix representing R), $\mathbf{W}_1$, $\mathbf{W}_2$, ..., $\mathbf{W}_n$, the last of which represents the transitive closure of R. Each matrix $\mathbf{W}_k$ comes from the matrix $\mathbf{W}_{k-1}$ in the following way. The $(i,j)^{\text{th}}$ entry of $\mathbf{W}_k$ is the "$\vee$" of the $(i,j)^{\text{th}}$ entry of $\mathbf{W}_{k-1}$ with the "$\wedge$" of the $(i,k)^{\text{th}}$ entry and the $(k,j)^{\text{th}}$ entry of $\mathbf{W}_{k-1}$. We will exhibit our solution by listing the matrices $\mathbf{W}_0$, $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{W}_3$, $\mathbf{W}_4$, in that order; $\mathbf{W}_4$ represents the answer. In each case $\mathbf{W}_0$ is the matrix of the given relation. To compute the next matrix in the solution, we need to compute it one entry at a time, using the equation just discussed (the "$\vee$" of the corresponding entry in the previous matrix with the "$\wedge$" of two entries in the old matrix), i.e., as i and j each go from 1 to 4, we need to write down the $(i,j)^{\text{th}}$ entry using this formula. Note that in computing $\mathbf{W}_k$ the k^{th} row and the k^{th} column are unchanged, but some of the entries in other rows and columns may change.

a) $\begin{bmatrix}0&1&0&0\\1&0&1&0\\0&0&0&1\\1&0&0&0\end{bmatrix}$ $\begin{bmatrix}0&1&0&0\\1&1&1&0\\0&0&0&1\\1&1&0&0\end{bmatrix}$ $\begin{bmatrix}1&1&1&0\\1&1&1&0\\0&0&0&1\\1&1&1&0\end{bmatrix}$ $\begin{bmatrix}1&1&1&1\\1&1&1&1\\0&0&0&1\\1&1&1&1\end{bmatrix}$ $\begin{bmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}$

b) $\begin{bmatrix}0&0&0&0\\1&0&1&0\\1&0&0&1\\1&0&1&0\end{bmatrix}$ $\begin{bmatrix}0&0&0&0\\1&0&1&0\\1&0&0&1\\1&0&1&0\end{bmatrix}$ $\begin{bmatrix}0&0&0&0\\1&0&1&0\\1&0&0&1\\1&0&1&0\end{bmatrix}$ $\begin{bmatrix}0&0&0&0\\1&0&1&1\\1&0&0&1\\1&0&1&1\end{bmatrix}$ $\begin{bmatrix}0&0&0&0\\1&0&1&1\\1&0&1&1\\1&0&1&1\end{bmatrix}$

c) $\begin{bmatrix}0&1&1&1\\0&0&1&1\\0&0&0&1\\0&0&0&0\end{bmatrix}$ $\begin{bmatrix}0&1&1&1\\0&0&1&1\\0&0&0&1\\0&0&0&0\end{bmatrix}$ $\begin{bmatrix}0&1&1&1\\0&0&1&1\\0&0&0&1\\0&0&0&0\end{bmatrix}$ $\begin{bmatrix}0&1&1&1\\0&0&1&1\\0&0&0&1\\0&0&0&0\end{bmatrix}$ $\begin{bmatrix}0&1&1&1\\0&0&1&1\\0&0&0&1\\0&0&0&0\end{bmatrix}$

Note that the relation was already transitive, so each matrix in the sequence was the same.

d) $\begin{bmatrix}1&0&0&1\\1&0&1&0\\1&1&0&1\\0&1&0&0\end{bmatrix}$ $\begin{bmatrix}1&0&0&1\\1&0&1&1\\1&1&0&1\\0&1&0&0\end{bmatrix}$ $\begin{bmatrix}1&0&0&1\\1&0&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}$ $\begin{bmatrix}1&0&0&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}$ $\begin{bmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{bmatrix}$

29. a) We need to include at least the transitive closure, which we can compute by Algorithm 1 or Algorithm 2 to be (in matrix form) $\begin{bmatrix}1&1&0&1\\0&0&0&0\\0&0&1&0\\1&1&0&1\end{bmatrix}$. All we need in addition is the pair $(2,2)$ in order to make the relation reflexive. Note that the result is still transitive (the addition of a pair (a,a) cannot make a transitive relation no longer transitive), so our answer is $\begin{bmatrix}1&1&0&1\\0&1&0&0\\0&0&1&0\\1&1&0&1\end{bmatrix}$.

b) The symmetric closure of the original relation is represented by $\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$. We need at least the transitive closure of this relation, namely $\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$. Since it is also symmetric, we are done. Note that it would not have been correct to find first the transitive closure of the original matrix and then make it symmetric, since the pair $(2,2)$ would be missing. What is going on here is that the transitive closure of a symmetric relation is still symmetric, but the symmetric closure of a transitive relation might not be transitive.

c) Since the answer to part **(b)** was already reflexive, it must be the answer to this part as well.

31. Algorithm 1 has a loop executed $O(n)$ times in which the primary operation is the Boolean product computation (the join operation is fast by comparison). If we can do the product in $O(n^{2.8})$ bit operations, then the number of bit operations in the entire algorithm is $O(n \cdot n^{2.8}) = O(n^{3.8})$. Since Algorithm 2 does not use the Boolean product, a fast Boolean product algorithm is irrelevant, so Algorithm 2 still requires $O(n^3)$ bit operations.

33. There are two ways to go. One approach is to take the output of Algorithm 1 as it stands and then make sure that all the pairs (a,a) are included by forming the join with the identity matrix (specifically set $\mathbf{B} := \mathbf{B} \vee \mathbf{I}_n$). See the discussion in Exercise 29a for the justification. The other approach is to insure the reflexivity at the beginning by initializing $\mathbf{A} := \mathbf{M}_r \vee \mathbf{I}_n$; if we do this, then only paths of length strictly less than n need to be looked at, so we can change the n in the loop to $n-1$.

35. **a)** No relation that contains R is not reflexive, since R already contains all the pairs $(0,0)$, $(1,1)$, and $(2,2)$. Therefore there is no "nonreflexive" closure of R.

b) Suppose S were the closure of R with respect to this property. Since R does not have an odd number of elements, $S \neq R$, so S must be a proper superset of R. Clearly S cannot have more than 5 elements, for if it did, then any subset of S consisting of R and one element of $S - R$ would be a proper subset of S with the property; this would violate the requirement that S be a subset of every superset of R with the property. Thus S must have exactly 5 elements. Let T be another superset of R with 5 elements (there are $9 - 4 = 5$ such sets in all). Thus T has the property, but S is not a subset of T. This contradicts the definition. Therefore our original assumption was faulty, and the closure does not exist.

SECTION 9.5 Equivalence Relations

This section is extremely important. If you do nothing else, do Exercise 9 and understand it, for it deals with the most common instances of equivalence relations. (See the comments in our solution below for some added insight.) Exercise 16 is interesting—it hints at what fractions really are (if understood properly) and perhaps helps to explain why children (and adults) usually have so much trouble with fractions: they really involve equivalence relations. Spend some time thinking about fractions in this context. (See also Writing Project 4 for this chapter.)

It is usually easier to understand equivalence relations in terms of the associated partition—it's a more concrete visual image. Thus make sure you understand exactly what Theorem 2 says. Look at Exercise 67 for the relationship between equivalence relations and closures.

1. In each case we need to check for reflexivity, symmetry, and transitivity.

a) This is an equivalence relation; it is easily seen to have all three properties. The equivalence classes all have just one element.

b) This relation is not reflexive since the pair $(1,1)$ is missing. It is also not transitive, since the pairs $(0,2)$ and $(2,3)$ are there, but not $(0,3)$.

c) This is an equivalence relation. The elements 1 and 2 are in the same equivalence class; 0 and 3 are each in their own equivalence class.

d) This relation is reflexive and symmetric, but it is not transitive. The pairs $(1,3)$ and $(3,2)$ are present, but not $(1,2)$.

e) This relation would be an equivalence relation were the pair $(2,1)$ present. As it is, its absence makes the relation neither symmetric nor transitive.

3. As in Exercise 1, we need to check for reflexivity, symmetry, and transitivity.

a) This is an equivalence relation, one of the general form that two things are considered equivalent if they have the same "something" (see Exercise 9 for a formalization of this idea). In this case the "something" is the value at 1.

b) This is not an equivalence relation because it is not transitive. Let $f(x) = 0$, $g(x) = x$, and $h(x) = 1$ for all $x \in \mathbf{Z}$. Then f is related to g since $f(0) = g(0)$, and g is related to h since $g(1) = h(1)$, but f is not related to h since they have no values in common. By inspection we see that this relation is reflexive and symmetric.

c) This relation has none of the three properties. It is not reflexive, since $f(x) - f(x) = 0 \neq 1$. It is not symmetric, since if $f(x) - g(x) = 1$, then $g(x) - f(x) = -1 \neq 1$. It is not transitive, since if $f(x) - g(x) = 1$ and $g(x) - h(x) = 1$, then $f(x) - h(x) = 2 \neq 1$.

d) This is an equivalence relation. Two functions are related here if they differ by a constant. It is clearly reflexive (the constant is 0). It is symmetric, since if $f(x) - g(x) = C$, then $g(x) - f(x) = -C$. It is transitive, since if $f(x) - g(x) = C_1$ and $g(x) - h(x) = C_2$, then $f(x) - h(x) = C_3$, where $C_3 = C_1 + C_2$ (add the first two equations).

e) This relation is not reflexive, since there are lots of functions f (for instance, $f(x) = x$) that do not have the property that $f(0) = f(1)$. It is symmetric by inspection (the roles of f and g are the same). It is not transitive. For instance, let $f(0) = g(1) = h(0) = 7$, and let $f(1) = g(0) = h(1) = 3$; fill in the remaining values arbitrarily. Then f and g are related, as are g and h, but f is not related to h since $7 \neq 3$.

5. Obviously there are many possible answers here. We can say that two buildings are equivalent if they were opened during the same year; an equivalence class consists of the set of buildings opened in a given year (as long as there was at least one building opened that year). For another example, we can define two buildings to be equivalent if they have the same number of stories; the equivalence classes are the set of 1-story buildings, the set of 2-story buildings, and so on (one class for each n for which there is at least one n-story building). In our third example, partition the set of all buildings into two classes—those in which you do have a class this semester and those in which you don't. (We assume that each of these is nonempty.) Every building in which you have a class is equivalent to every building in which you have a class (including itself), and every building in which you don't have a class is equivalent to every building in which you don't have a class (including itself).

7. Two propositions are equivalent if their truth tables are identical. This relation is reflexive, since the truth table of a proposition is identical to itself. It is symmetric, since if p and q have the same truth table, then q and p have the same truth table. There is one technical point about transitivity that should be noted. We need to assume that the truth tables, as we consider them for three propositions p, q, and r, have the same

atomic variables in them. If we make this assumption (and it cannot hurt to do so, since adding information about extra variables that do not appear in a pair of propositions does not change the truth value of the propositions), then we argue in the usual way: if p and q have identical truth tables, and if q and r have identical truth tables, then p and r have that same common truth table. The proposition **T** is always true; therefore the equivalence class for this proposition consists of all propositions that are always true, no matter what truth values the atomic variables have. Recall that we call such a proposition a tautology. Therefore the equivalence class of **T** is the set of all tautologies. Similarly, the equivalence class of **F** is the set of all contradictions.

9. This is an important exercise, since very many equivalence relations are of this form. (In fact, all of them are—see Exercise 10. A relation defined by a condition of the form "x and y are equivalent if and only if they have the same ..." is an equivalence relation. The function f here tells what about x and y are "the same.")

a) This relation is reflexive, since obviously $f(x) = f(x)$ for all $x \in A$. It is symmetric, since if $f(x) = f(y)$, then $f(y) = f(x)$ (this is one of the fundamental properties of equality). It is transitive, since if $f(x) = f(y)$ and $f(y) = f(z)$, then $f(x) = f(z)$ (this is another fundamental property of equality).

b) The equivalence class of x is the set of all $y \in A$ such that $f(y) = f(x)$. This is by definition just the inverse image of $f(x)$. Thus the equivalence classes are precisely the sets $f^{-1}(b)$ for every b in the range of f.

11. This follows from Exercise 9, where f is the function that takes a bit string of length 3 or more to its first 3 bits.

13. This follows from Exercise 9, where f is the function that takes a bit string of length 3 or more to the ordered pair (b_1, b_3), where b_1 is the first bit of the string and b_3 is the third bit of the string. Two bit strings agree on their first and third bits if and only if the corresponding ordered pairs for these two strings are equal ordered pairs.

15. By algebra, the given condition is the same as the condition that $f((a,b)) = f((c,d))$, where $f((x,y)) = x - y$. Therefore by Exercise 9 this is an equivalence relation. If we want a more explicit proof, we can argue as follows. For reflexivity, $((a,b),(a,b)) \in R$ because $a + b = b + a$. For symmetry, $((a,b),(c,d)) \in R$ if and only if $a + d = b + c$, which is equivalent to $c + b = d + a$, which is true if and only if $((c,d),(a,b)) \in R$. For transitivity, suppose $((a,b),(c,d)) \in R$ and $((c,d),(e,f)) \in R$. Thus we have $a + d = b + c$ and $c + e = d + f$. Adding, we obtain $a + d + c + e = b + c + d + f$. Simplifying, we have $a + e = b + f$, which tells us that $((a,b),(e,f)) \in R$.

17. **a)** This follows from Exercise 9, where the function f from the set of differentiable functions (from **R** to **R**) to the set of functions (from **R** to **R**) is the differentiation operator—i.e., f of a function g is the function g'. The best way to think about this is that any relation defined by a statement of the form "a and b are equivalent if they have the same whatever" is an equivalence relation. Here "whatever" is "derivative"; in the general situation of Exercise 9, "whatever" is "function value under f."

b) We are asking for all functions that have the same derivative that the function $f(x) = x^2$ has, i.e., all functions of x whose derivative is $2x$. In other words, we are asking for the general antiderivative of $2x$, and we know that $\int 2x = x^2 + C$, where C is any constant. Therefore the functions in the same equivalence class as $f(x) = x^2$ are all the functions of the form $g(x) = x^2 + C$ for some constant C. Indefinite integrals in calculus, then, give equivalence classes of functions as answers, not just functions.

19. This follows from Exercise 9, where the function f from the set of all URLs to the set of all Web pages is the function that assigns to each URL the Web page for that URL.

21. We need to observe whether the relation is reflexive (there is a loop at each vertex), symmetric (every edge that appears is accompanied by its antiparallel mate—an edge involving the same two vertices but pointing in the opposite direction), and transitive (paths of length 2 are accompanied by the path of length 1—i.e., edge—between the same two vertices in the same direction). We see that this relation is not transitive, since the edges (c, d) and (d, c) are missing.

23. As in Exercise 21, this relation is not transitive, since several required edges are missing (such as (a, c)).

25. This follows from Exercise 9, with f being the function from bit strings to nonnegative integers given by $f(s) =$the number of 1's in s.

27. Only parts **(a)** and **(b)** are relevant here, since the others are not equivalence relations.
a) An equivalence class is the set of all people who are the same age. (To really identify the equivalence class and the equivalence relation itself, one would need to specify exactly what one meant by "the same age." For example, we could define two people to be the same age if their official dates of birth were identical. In that case, everybody born on April 25, 1948, for example, would constitute one equivalence class.)
b) For each pair (m, f) of a man and a woman, the set of offspring of their union, if nonempty, is an equivalence class. In many cases, then, an equivalence class consists of all the children in a nuclear family with children. (In real life, of course, this is complicated by such things as divorce and remarriage.)

29. The equivalence class of 011 is the set of all bit strings that are related to 011, namely the set of all bit strings that have the same number of 1's as 011. In other words, it is the (infinite) set of all bit strings with exactly 2 1's: $\{11, 110, 101, 011, 1100, 1010, 1001, \ldots\}$.

31. Since two strings are related if they agree beyond their first 3 bits, the equivalence class of a bit string $xyzt$, where x, y, and z are bits, and t is a bit string, is the set of all bit strings of the form $x'y'z't$, where x', y', and z' are any bits.
a) the set of all bit strings of length 3 (take $t = \lambda$ in the formulation given above)
b) the set of all bit strings of length 4 that end with a 1
c) the set of all bit strings of length 5 that end 11
d) the set of all bit strings of length 8 that end 10101

33. This is like Example 15. Each bit string of length less than 4 is in an equivalence class by itself ($[\lambda]_{R_4} = \{\lambda\}$, $[0]_{R_4} = \{0\}$, $[1]_{R_4} = \{1\}$, $[00]_{R_4} = \{00\}$, $[01]_{R_4} = \{01\}$, $\ldots$, $[111]_{R_4} = \{111\}$). This accounts for $1+2+4+8 = 15$ equivalence classes. The remaining 16 equivalence classes are determined by the bit strings of length 4:

$$[0000]_{R_4} = \{0000, 00000, 00001, 000000, 000001, 000010, 000011, 0000000, \ldots\}$$
$$[0001]_{R_4} = \{0001, 00010, 00011, 000100, 000101, 000110, 000111, 0001000, \ldots\}$$
$$[0010]_{R_4} = \{0010, 00100, 00101, 001000, 001001, 001010, 001011, 0010000, \ldots\}$$
$$\vdots$$
$$[1111]_{R_4} = \{1111, 11110, 11111, 111100, 111101, 111110, 111111, 1111000, \ldots\}$$

35. We have by definition that $[n]_5 = \{\, i \mid i \equiv n \pmod 5 \,\}$.
a) $[2]_5 = \{\, i \mid i \equiv 2 \pmod 5 \,\} = \{\ldots, -8, -3, 2, 7, 12, \ldots\}$
b) $[3]_5 = \{\, i \mid i \equiv 3 \pmod 5 \,\} = \{\ldots, -7, -2, 3, 8, 13, \ldots\}$
c) $[6]_5 = \{\, i \mid i \equiv 6 \pmod 5 \,\} = \{\ldots, -9, -4, 1, 6, 11, \ldots\}$
d) $[-3]_5 = \{\, i \mid i \equiv -3 \pmod 5 \,\} = \{\ldots, -8, -3, 2, 7, 12, \ldots\}$ (the same as $[2]_5$)

37. This is very similar to Example 14. There are 6 equivalence classes, namely

$$[0]_6 = \{\ldots, -12, -6, 0, 6, 12, \ldots\},$$
$$[1]_6 = \{\ldots, -11, -5, 1, 7, 13, \ldots\},$$
$$[2]_6 = \{\ldots, -10, -4, 2, 8, 14, \ldots\},$$
$$[3]_6 = \{\ldots, -9, -3, 3, 9, 15, \ldots\},$$
$$[4]_6 = \{\ldots, -8, -2, 4, 10, 16, \ldots\},$$
$$[5]_6 = \{\ldots, -7, -1, 5, 11, 17, \ldots\}.$$

Another way to describe this collection is to say that it is the collection of sets $\{\, 6n + k \mid n \in \mathbf{Z} \,\}$ for $k = 0, 1, 2, 3, 4, 5$.

39. **a)** We observed in the solution to Exercise 15 that (a, b) is equivalent to (c, d) if $a - b = c - d$. Thus because $1 - 2 = -1$, we have $[(1, 2)] = \{\, (a, b) \mid a - b = -1 \,\} = \{(1, 2), (3, 4), (4, 5), (5, 6), \ldots\}$.

b) Since the equivalence class of (a, b) is entirely *determined* by the integer $a - b$, which can be negative, positive, or zero, we can interpret the equivalences classes as *being* the integers. This is a standard way to *define* the integers once we have defined the whole numbers.

41. The sets in a partition must be nonempty, pairwise disjoint, and have as their union all of the underlying set.

a) This is not a partition, since the sets are not pairwise disjoint (the elements 2 and 4 each appear in two of the sets).

b) This is a partition. **c)** This is a partition.

d) This is not a partition, since none of the sets includes the element 3.

43. In each case, we need to see that the collection of subsets satisfy three conditions: they are nonempty, they are pairwise disjoint, and their union is the entire set of 256 bit strings of length 8.

a) This is a partition, since strings must begin either 1 or 0, and those that begin 0 must continue with either 0 or 1 in their second position. It is clear that the three subsets satisfy the conditions.

b) This is not a partition, since these subsets are not pairwise disjoint. The string 00000001, for example, contains both 00 and 01.

c) This is clearly a partition. Each of these four subsets contains 64 bit strings, and no two of them overlap.

d) This is not a partition, because the union of these subsets is not the entire set. For example, the string 00000010 is in none of the subsets.

e) This is a partition. Each bit string contains some number of 1's. This number can be identified in exactly one way as of the form $3k$, the form $3k + 1$, or the form $3k + 2$, where k is a nonnegative integer; it really is just looking at the equivalence classes of the number of 1's modulo 3.

45. In each case, we need to see that the collection of subsets satisfy three conditions: they are nonempty, they are pairwise disjoint, and their union is the entire set $\mathbf{Z} \times \mathbf{Z}$.

a) This is not a partition, since the subsets are not pairwise disjoint. The pair $(2, 3)$, for example, is in both of the first two subsets listed.

b) This is a partition. Every pair satisfies exactly one of the conditions listed about the parity of x and y, and clearly these subsets are nonempty.

c) This is not a partition, since the subsets are not pairwise disjoint. The pair $(2, 3)$, for example, is in both of the first two subsets listed. Also, $(0, 0)$ is in none of the subsets.

d) This is a partition. Every pair satisfies exactly one of the conditions listed about the divisibility of x and y by 3, and clearly these subsets are nonempty.

e) This is a partition. Every pair satisfies exactly one of the conditions listed about the positiveness of x and y, and clearly these subsets are nonempty.

f) This is not a partition, because the union of these subsets is not all of $\mathbf{Z} \times \mathbf{Z}$. In particular, $(0,0)$ is in none of the parts.

47. In each case, we need to list all the pairs we can where both coordinates are chosen from the same subset. We should proceed in an organized fashion, listing all the pairs corresponding to each part of the partition.

a) $\{(0,0),(1,1),(1,2),(2,1),(2,2),(3,3),(3,4),(3,5),(4,3),(4,4),(4,5),(5,3),(5,4),(5,5)\}$

b) $\{(0,0),(0,1),(1,0),(1,1),(2,2),(2,3),(3,2),(3,3),(4,4),(4,5),(5,4),(5,5)\}$

c) $\{(0,0),\ (0,1),\ (0,2),\ (1,0),\ (1,1),\ (1,2),\ (2,0),\ (2,1),\ (2,2),\ (3,3),\ (3,4),\ (3,5),\ (4,3),\ (4,4),\ (4,5),\ (5,3),\ (5,4),(5,5)\}$

d) $\{(0,0),(1,1),(2,2),(3,3),(4,4),(5,5)\}$

49. We need to show that every equivalence class modulo 6 is contained in an equivalence class modulo 3. We claim that in fact, for each $n \in \mathbf{Z}$, $[n]_6 \subseteq [n]_3$. To see this suppose that $m \in [n]_6$. This means that $m \equiv n \pmod 6$, i.e., that $m - n$ is a multiple of 6. Then perforce $m - n$ is a multiple of 3, so $m \equiv n \pmod 3$, which means that $m \in [n]_3$.

51. By the definition given in the preamble to Exercise 49, we need to show that every set in the first partition is a subset of some set in the second partition. Let A be a set in the first partition. So A is the set of all bit strings of length 16 that agree on their last eight bits. Pick a particular element x of A, and suppose that the last four bits of x are *abcd*. Then the set of all bit strings of length 16 whose last four bits are *abcd* is one of the sets in the second partition, and clearly every string in A is in that set, since every string in A agrees with x on the last eight bits, and therefore perforce agrees on the last four bits.

53. We are asked to show that every equivalence class for R_{31} is a subset of some equivalence class for R_8. Let $[x]_{R_{31}}$ be an arbitrary equivalence class for R_{31}. We claim that $[x]_{R_{31}} \subseteq [x]_{R_8}$; proving this claim finishes the proof. To show that one set is a subset of another set, we choose an arbitrary element y in the first set and show that it is also an element of the second set. In this case since $y \in [x]_{R_{31}}$, we know that y is equivalent to x under R_{31}, that is, that either $y = x$ or y and x are each at least 31 characters long and agree on their first 31 characters. Because strings that are at least 31 characters long and agree on their first 31 characters perforce are at least 8 characters long and on their first 8 characters, we know that either $y = x$ or y and x are each at least 8 characters long and agree on their first 8 characters. This means that y is equivalent to x under R_8, that is, that $y \in [x]_{R_8}$.

55. We need first to make the relation symmetric, so we add the pairs (b,a), (c,a), and (e,d). Then we need to make it transitive, so we add the pairs (b,c), (c,b), (a,a), (b,b), (c,c), (d,d), and (e,e). (In other words, we formed the transitive closure of the symmetric closure of the original relation.) It happens that we have already achieved reflexivity, so we are done; if there had been some pairs (x,x) missing at this point, we would have added them as well. Thus the desired equivalence relation is the one consisting of the original 3 pairs and the 10 we have added. There are two equivalence classes, $\{a,b,c\}$ and $\{d,e\}$.

57. **a)** The equivalence class of 1 is the set of all real numbers that differ from 1 by an integer. Obviously this is the set of all integers.

b) The equivalence class of $1/2$ is the set of all real numbers that differ from $1/2$ by an integer, namely $1/2$, $3/2$, $5/2$, etc., and $-1/2$, $-3/2$, etc. These are often called **half-integers**. We could write this set as $\{\,(2n+1)/2 \mid n \in \mathbf{Z}\,\}$, among other ways.

59. This problem actually deals with a branch of mathematics called group theory; the object being studied here is related to a certain dihedral group. If this fascinates you, you might want to take a course with a title like Abstract Algebra or Modern Algebra, in which such things are studied in depth.

In order to have a way to talk about specific colorings, let us agree that a sequence of length four, each element of which is either r or b, represents a coloring of the 2×2 checkerboard, where the first letter denotes the color of the upper left square, the second letter denotes the color of the upper right square, the third letter denotes the color of the lower left square, and the fourth letter denotes the color of the lower right square. For example, the board in which every square is red except the upper right would be represented by *rbrr*. There are really only four different rotations, since after the rotation we need to end up with another checkerboard (and we can assume that the edges of the board are horizontal and vertical). If we rotate our sample coloring 90° clockwise, then we obtain the coloring *rrrb*; if we rotate it 180°, then we obtain the coloring *rrbr*; if we rotate it 270° clockwise (or 90° counterclockwise), then we obtain the coloring *brrr*; and if we rotate it 360° clockwise (or 0°—i.e., not at all), then we obtain the coloring *rbrr* itself back. Note also that some colorings are *invariant* (i.e., unchanged) under rotations in addition to the 360° one; for example, *bbbb* is invariant under all rotations, and *brrb* is invariant under a 180° rotation. Similarly there are four reflections: around the center vertical axis of the board, around the center horizontal axis, around the lower-left-to-upper-right diagonal, and around the lower-right-to-upper-left diagonal. For example, applying the vertical axis reflection to *rrbb* yields itself, while applying the lower-left-to-upper-right diagonal reflection results in *brbr*.

The definition of equivalence for this problem makes the proof rather messy, since both rotations and reflections are involved, and it is required that we reduce everything to just one or two operations. In fact, we claim that there are only eight possible motions of this square: clockwise rotations of 0°, 90°, 180°, or 270°, and reflections through the vertical, horizontal, lower-left-to-upper-right, and lower-right-to-upper-left diagonals. To verify this, we must show that the composition of every two of these operations is again an operation in our list. Below is the "group table" that shows this, where we use the symbols $r0$, $r90$, $r180$, $r270$, fv, fh, fp, and fn for these operations, respectively. (The mnemonic is that r stands for "rotation," f stands for "flip," and v, h, p, and n stand for "vertical," "horizontal", "positive-sloping," and "negative-sloping," respectively.) It is read just like a multiplication table, with the operation $\circ$ meaning "followed by." For example, if we first perform $r90$ and then perform fh, then we get the same result as if we had just performed fp (try it!).

$\circ$	$r0$	$r90$	$r180$	$r270$	fv	fh	fp	fn
$r0$	$r0$	$r90$	$r180$	$r270$	fv	fh	fp	fn
$r90$	$r90$	$r180$	$r270$	$r0$	fn	fp	fv	fh
$r180$	$r180$	$r270$	$r0$	$r90$	fh	fv	fn	fp
$r270$	$r270$	$r0$	$r90$	$r180$	fp	fn	fh	fv
fv	fv	fp	fh	fn	$r0$	$r180$	$r90$	$r270$
fh	fh	fn	fv	fp	$r180$	$r0$	$r270$	$r90$
fp	fp	fh	fn	fv	$r270$	$r90$	$r0$	$r180$
fn	fn	fv	fp	fh	$r90$	$r270$	$r180$	$r0$

So the result of this computation is that we can consider only these eight moves, and not have to worry about combinations of them—every combination of moves equals just one of these eight.

a) To show reflexivity, we note that every coloring can be obtained from itself via a 0° rotation. In technical terms, the 0° rotation is the *identity element* of our group. To show symmetry, we need to observe that rotations and reflections have inverses: If C_1 comes from C_2 via a rotation of $n°$ clockwise, then C_2 comes from C_1 via a rotation of $n°$ counterclockwise (or equivalently, via a rotation of $(360 - n)°$ clockwise); and every reflection applied twice brings us back to the position (and therefore coloring) we began with.

And transitivity follows from the fact that the composition of two of these operations is again one of these operations.

b) The equivalence classes are represented by colorings that are truly distinct, in the sense of not being obtainable from each other via these operations. Let us list them. Clearly there is just one coloring using four red squares, and so just one equivalence class, $[rrrr]$. Similarly there is only one using four blues, $[bbbb]$. There is also just one equivalence class of colorings using three reds and one blue, since no matter which corner the single blue occupies in such a coloring, we can rotate to put the blue in any other corner. Thus our third and fourth equivalence classes are $[rrrb]$ and $[bbbr]$. Note that each of them contains four colorings. (For example, $[rrrb] = \{rrrb, rrbr, rbrr, brrr\}$.) This leaves only the colorings with two reds and two blues to consider. In every such coloring, either the red squares are adjacent (i.e., share a common edge), such as in $bbrr$, or they are not (e.g., $brrb$). Clearly the red squares are adjacent if and only if the blue ones are, since the only pairs of nonadjacent squares are (lower-left,upper-right) and (upper-left,lower-right). It is equally clear that there are only two colorings in which the red squares are not adjacent, namely $rbbr$ and $brrb$, and they are equivalent via a 90° rotation (among other transformations). So our fifth equivalence class is $[rbbr] = \{rbbr, brrb\}$. Finally, there is only one more equivalence class, and it contains the remaining four colorings (in which the two red squares are adjacent and the two blue squares are adjacent), namely $\{rrbb, brbr, bbrr, rbrb\}$, since each of these can be obtained from each of the others by a rotation. In summary we have partitioned the set of $2^4 = 16$ colorings (i.e., r-b strings of length four) into six equivalence classes, two of which have cardinality one, three of which have cardinality four, and one of which has cardinality two.

One final comment. We saw in the solution to part **(b)** that only rotations are needed to show the equivalence of every pair of equivalent colorings using just red and blue. This means that we are actually dealing with just part of the dihedral group here. If more colors had been used, then we would have needed to use the reflections as well. A complete discussion would get us into Pólya's theory of enumeration, which is studied in advanced combinatorics classes.

61. It is easier to write down a partition than it is to list the pairs in an equivalence relation, so we will answer the question using this notation. Let the set be $\{1, 2, 3\}$. We want to write down all possible partitions of this set. One partition is just $\{\{1, 2, 3\}\}$, i.e., having just one set (this corresponds to the equivalence relation in which every pair of elements are related). At the other extreme, there is the partition $\{\{1\}, \{2\}, \{3\}\}$, which corresponds to the equality relation (each x is related only to itself). The only other way to split up the elements of this set is into a set with two elements and a set with one element, and there are clearly three ways to do this, depending on which element we decide to put in the set by itself. Thus we get the partitions (pay attention to the punctuation!) $\{\{1, 2\}, \{3\}\}$, $\{\{1, 3\}, \{2\}\}$, and $\{\{2, 3\}, \{1\}\}$. If we wished to list the ordered pairs, we could; for example, the relation corresponding to $\{\{2, 3\}, \{1\}\}$ is $\{(2, 2), (2, 3), (3, 2), (3, 3), (1, 1)\}$. We found five partitions, so the answer to the question is 5.

63. We do get an equivalence relation. The issue is whether the relation formed in this way is reflexive, transitive and symmetric. It is clearly reflexive, since we included all the pairs (a, a) at the outset. It is clearly transitive, since the last thing we did was to form the transitive closure. It is symmetric by Exercise 23 in Section 9.4.

65. We end up with the relation R that we started with. Two elements are related if they are in the same set of the partition, but the partition is made up of the equivalence classes of R, so two elements are related precisely if they are related in R.

67. We make use of Exercise 63. Given the relation R, we first form the reflexive closure R' of R by adding to R each pair (a, a) that is not already there. Next we form the symmetric closure R'' of R', by adding, for each pair $(a, b) \in R'$ the pair (b, a) if it is not already there. Finally we apply Warshall's algorithm (or

Algorithm 1) from Section 9.4 to form the transitive closure of R''. This is the smallest equivalence relation containing R.

69. The exercise asks us to compute $p(n)$ for $n = 0, 1, 2, \ldots, 10$. In doing this we will use the recurrence relation, building on what we have already computed (namely $p(n-j-1)$, noting that $n-j-1 < n$), as well as using the binomial coefficients $C(n-1, j) = \dfrac{(n-1)!}{j!(n-1-j)!}$. We organize our computation in the obvious way, using the formula in Exercise 68.

$$p(0) = 1 \quad \text{(the initial condition)}$$

$$p(1) = C(0,0)p(0) = 1 \cdot 1 = 1$$

$$p(2) = C(1,0)p(1) + C(1,1)p(0) = 1 \cdot 1 + 1 \cdot 1 = 2$$

$$p(3) = C(2,0)p(2) + C(2,1)p(1) + C(2,2)p(0) = 1 \cdot 2 + 2 \cdot 1 + 1 \cdot 1 = 5$$

$$p(4) = C(3,0)p(3) + C(3,1)p(2) + C(3,2)p(1) + C(3,3)p(0) = 1 \cdot 5 + 3 \cdot 2 + 3 \cdot 1 + 1 \cdot 1 = 15$$

$$\begin{aligned} p(5) &= C(4,0)p(4) + C(4,1)p(3) + C(4,2)p(2) + C(4,3)p(1) + C(4,4)p(0) \\ &= 1 \cdot 15 + 4 \cdot 5 + 6 \cdot 2 + 4 \cdot 1 + 1 \cdot 1 = 52 \end{aligned}$$

$$\begin{aligned} p(6) &= C(5,0)p(5) + C(5,1)p(4) + C(5,2)p(3) + C(5,3)p(2) + C(5,4)p(1) + C(5,5)p(0) \\ &= 1 \cdot 52 + 5 \cdot 15 + 10 \cdot 5 + 10 \cdot 2 + 5 \cdot 1 + 1 \cdot 1 = 203 \end{aligned}$$

$$\begin{aligned} p(7) &= C(6,0)p(6) + C(6,1)p(5) + C(6,2)p(4) + C(6,3)p(3) + C(6,4)p(2) + C(6,5)p(1) + C(6,6)p(0) \\ &= 1 \cdot 203 + 6 \cdot 52 + 15 \cdot 15 + 20 \cdot 5 + 15 \cdot 2 + 6 \cdot 1 + 1 \cdot 1 = 877 \end{aligned}$$

$$\begin{aligned} p(8) &= C(7,0)p(7) + C(7,1)p(6) + C(7,2)p(5) + C(7,3)p(4) + C(7,4)p(3) + C(7,5)p(2) \\ &\quad + C(7,6)p(1) + C(7,7)p(0) \\ &= 1 \cdot 877 + 7 \cdot 203 + 21 \cdot 52 + 35 \cdot 15 + 35 \cdot 5 + 21 \cdot 2 + 7 \cdot 1 + 1 \cdot 1 = 4140 \end{aligned}$$

$$\begin{aligned} p(9) &= C(8,0)p(8) + C(8,1)p(7) + C(8,2)p(6) + C(8,3)p(5) + C(8,4)p(4) + C(8,5)p(3) \\ &\quad + C(8,6)p(2) + C(8,7)p(1) + C(8,8)p(0) \\ &= 1 \cdot 4140 + 8 \cdot 877 + 28 \cdot 203 + 56 \cdot 52 + 70 \cdot 15 + 56 \cdot 5 + 28 \cdot 2 + 8 \cdot 1 + 1 \cdot 1 = 21147 \end{aligned}$$

$$\begin{aligned} p(10) &= C(9,0)p(9) + C(9,1)p(8) + C(9,2)p(7) + C(9,3)p(6) + C(9,4)p(5) + C(9,5)p(4) \\ &\quad + C(9,6)p(3) + C(9,7)p(2) + C(9,8)p(1) + C(9,9)p(0) \\ &= 1 \cdot 21147 + 9 \cdot 4140 + 36 \cdot 877 + 84 \cdot 203 + 126 \cdot 52 \\ &\quad + 126 \cdot 15 + 84 \cdot 5 + 36 \cdot 2 + 9 \cdot 1 + 1 \cdot 1 = 115975 \end{aligned}$$

SECTION 9.6 Partial Orderings

Partial orderings (or "partial orders"—the two phrases are used interchangeably) rival equivalence relations in importance in mathematics and computer science. Again, try to concentrate on the visual image—in this case the Hasse diagram. Play around with different posets to become familiar with the different possibilities; not all posets have to look like the less than or equal relation on the integers. Exercises 32 and 33 are important, and they are not difficult if you pay careful attention to the definitions.

1. The question in each case is whether the relation is reflexive, antisymmetric, and transitive. Suppose the relation is called R.

a) Clearly this relation is reflexive because each of 0, 1, 2, and 3 is related to itself. The relation is also antisymmetric, because the only way for a to be related to b is for a to equal b. Similarly, the relation is transitive, because if a is related to b, and b is related to c, then necessarily $a = b = c$ so a is related to c (because the relation is reflexive). This is just the equality relation on $\{0, 1, 2, 3\}$; more generally, the equality relation on any set satisfies all three conditions and is therefore a partial ordering. (It is the smallest partial ordering; reflexivity insures that every partial ordering contains at least all the pairs (a, a).)

b) This is not a partial ordering, because although the relation is reflexive, it is not antisymmetric (we have $2\,R\,3$ and $3\,R\,2$, but $2 \neq 3$), and not transitive ($3\,R\,2$ and $2\,R\,0$, but 3 is not related to 0).

c) This is a partial ordering, because it is clearly reflexive; is antisymmetric (we just need to note that $(1, 2)$ is the only pair in the relation with unequal components); and is transitive (for the same reason).

d) This is a partial ordering because it is the "less than or equal to" relation on $\{1, 2, 3\}$ together with the isolated point 0.

e) This is not a partial ordering. The relation is clearly reflexive, but it is not antisymmetric ($0\,R\,1$ and $1\,R\,0$, but $0 \neq 1$) and not transitive ($2\,R\,0$ and $0\,R\,1$, but 2 is not related to 1).

3. The question in each case is whether the relation is reflexive, antisymmetric, and transitive.

a) Since nobody is taller than himself, this relation is not reflexive so (S, R) cannot be a poset.

b) To be not taller means to be exactly the same height or shorter. Two different people x and y could have the same height, in which case $x\,R\,y$ and $y\,R\,x$ but $x \neq y$, so R is not antisymmetric and this is not a poset.

c) This is a poset. The equality clause in the definition of R guarantees that R is reflexive. To check antisymmetry and transitivity it suffices to consider unequal elements (these rules hold for equal elements trivially). If a is an ancestor of b, then b cannot be an ancestor of a (for one thing, an ancestor needs to be born before any descendant), so the relation is vacuously antisymmetric. If a is an ancestor of b, and b is an ancestor of c, then by the way "ancestor" is defined, we know that a is an ancestor of b; thus R is transitive.

d) This relation is not antisymmetric. Let a and b be any two distinct friends of yours. Then $a\,R\,b$ and $b\,R\,a$, but $a \neq b$.

5. The question in each case is whether the relation is reflexive, antisymmetric, and transitive.

a) The equality relation on any set satisfies all three conditions and is therefore a partial partial ordering. (It is the smallest partial partial ordering; reflexivity insures that every partial order contains at least all the pairs (a, a).)

b) This is not a poset, since the relation is not reflexive, not antisymmetric, and not transitive (the absence of one of these properties would have been enough to give a negative answer).

c) This is a poset, as explained in Example 1.

d) This is not a poset. The relation is not reflexive, since it is not true, for instance, that $2 \not| 2$. (It also is not antisymmetric and not transitive.)

7. **a)** This relation is $\{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (3, 3)\}$. It is not antisymmetric because $(1, 2)$ and $(2, 1)$ are both in the relation, but $1 \neq 2$. We can see this visually by the pair of 1's symmetrically placed around the main diagonal at positions $(1, 2)$ and $(2, 1)$. Therefore this matrix does not represent a partial order.

b) This matrix represents a partial order. Reflexivity is clear. The only other pairs in the relation are $(1, 2)$ and $(1, 3)$, and clearly neither can be part of a counterexample to antisymmetry or transitivity.

c) A little trial and error shows that this relation is not transitive ($(4, 1)$ and $(1, 3)$ are present, but not $(4, 3)$) and therefore not a partial order.

9. This relation is not transitive (there are arrows from a to b and from b to d, but there is no arrow from a to d), so it is not a partial order.

11. This relation is a partial order, since it has all three properties—it is reflexive (there is an arrow at each point), antisymmetric (there are no pairs of arrows going in opposite directions between two different points), and transitive (there is no missing arrow from some x to some z when there were arrows from x to y and y to z).

13. The dual of a poset is the poset with the same underlying set and with the relation defined by declaring a related to b if and only if $b \preceq a$ in the given poset.

a) The dual relation to $\leq$ is $\geq$, so the dual poset is $(\{0,1,2\}, \geq)$. Explicitly it is the set $\{(0,0),(1,0),(1,1),(2,0),(2,1),(2,2)\}$.

b) The dual relation to $\geq$ is $\leq$, so the dual poset is $(\mathbf{Z}, \leq)$.

c) The dual relation to $\supseteq$ is $\subseteq$, so the dual poset is $(P(\mathbf{Z}), \subseteq)$.

d) There is no symbol generally used for the "is a multiple of" relation, which is the dual to the "divides" relation in this part of the exercise. If we let R be the relation such that aRb if and only if $b \mid a$, then the answer can be written $(\mathbf{Z}^+, R)$.

15. We need to find elements such that the relation holds in neither direction between them. The answers we give are not the only ones possible.

a) One such pair is $\{1\}$ and $\{2\}$. These are both subsets of $\{0,1,2\}$, so they are in the poset, but neither is a subset of the other.

b) Neither 6 nor 8 divides the other, so they are incomparable.

17. We find the first coordinate (from left to right) at which the tuples differ and place first the tuple with the smaller value in that coordinate.

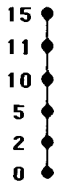
a) Since $1 = 1$ in the first coordinate, but $1 < 2$ in the second coordinate, $(1,1,2) < (1,2,1)$.

b) The first two coordinates agree, but $2 < 3$ in the third, so $(0,1,2,3) < (0,1,3,2)$.

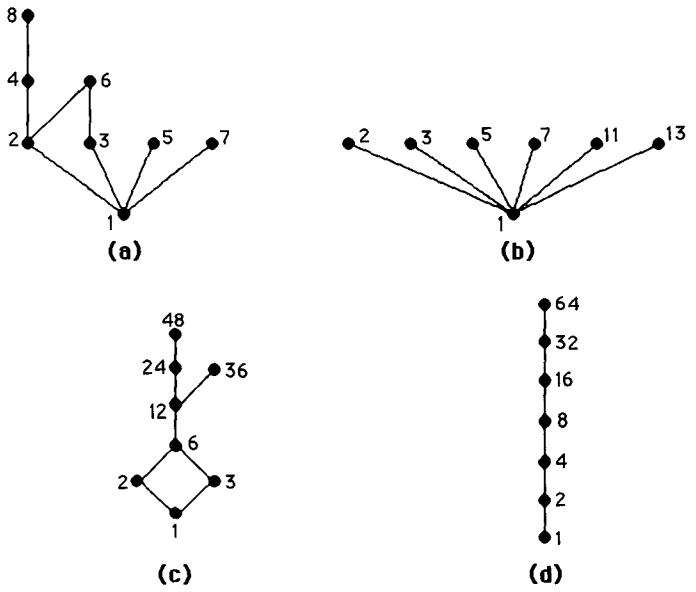
c) Since $0 < 1$ in the first coordinate, $(0,1,1,1,0) < (1,0,1,0,1)$.

19. All the strings that begin with 0 precede all those that begin with 1. The 0 comes first. Next comes 0001, which begins with three 0's, then 001, which begins with two 0's. Among the strings that begin 01, the order is $01 < 010 < 0101 < 011$. Putting this all together, we have $0 < 0001 < 001 < 01 < 010 < 0101 < 011 < 11$.

21. This is a totally ordered set, so the Hasse diagram is linear.



23. We put x above y if y divides x. We draw a line between x and y, where y divides x, if there is no number z in our set, other than x or y, such that $y \mid z \wedge z \mid x$. Note that in part **(b)** the numbers other than 1 are all (relatively) prime, so the Hasse diagram is short and wide, whereas in part **(d)** the numbers all divide one another, so the Hasse diagram is tall and narrow.



25. We need to include every pair (x, y) for which we can find a path going upward in the diagram from x to y. We also need to include all the reflexive pairs (x, x). Therefore the relation is the following set of pairs: $\{(a,a),(a,b),(a,c),(a,d),(b,b),(b,c),(b,d),(c,c),(d,d)\}$.

27. The procedure is the same as in Exercise 25: $\{(a,a),(a,d),(a,e),(a,f),(a,g),(b,b),(b,d),(b,e),(b,f),(b,g),(c,c),(c,d),(c,e),(c,f),(c,g),(d,d),(e,e),(f,f),(g,d),(g,e),(g,f),(g,g)\}$.

29. In this problem $X \preceq Y$ when $X \subseteq Y$. For (X, Y) to be in the covering relation, we need X to be a proper subset of Y but we also must have no subset strictly between X and Y. For example, $(\{a\},\{a,b,c\})$ is not in the covering relation, since $\{a\} \subset \{a,b\}$ and $\{a,b\} \subset \{a,b,c\}$. With this understanding it is easy to list the pairs in the covering relation: $(\emptyset,\{a\})$, $(\emptyset,\{b\})$, $(\emptyset,\{c\})$, $(\{a\},\{a,b\})$, $(\{a\},\{a,c\})$, $(\{b\},\{a,b\})$, $(\{b\},\{b,c\})$, $(\{c\},\{a,c\})$, $(\{c\},\{b,c\})$, $(\{a,b\},\{a,b,c\})$, $(\{a,c\},\{a,b,c\})$, and $(\{b,c\},\{a,b,c\})$.

31. Let $(S, \preceq)$ be a finite poset. We claim that this poset is just the reflexive transitive closure of its covering relation. Suppose that (a, b) is in the reflexive transitive closure of the covering relation. Then either $a = b$ or $a \prec b$ (in which cases certainly $a \preceq b$) or else there is a sequence $a \prec a_1 \prec a_2 \prec \cdots \prec a_n \prec b$, in which case again $a \preceq b$, by the transitivity of $\preceq$. Conversely, suppose that $a \preceq b$. If $a = b$, then (a, b) is certainly in the reflexive transitive closure of the covering relation. If $a \prec b$ and there is no z such that $a \prec z \prec b$, then $(a, b,)$ is in the covering relation and again therefore in its reflexive transitive closure. Otherwise, let $a \prec a_1 \prec a_2 \prec \cdots \prec a_n \prec b$ be a longest possible sequence of this form; since the poset is finite, there must be such a longest sequence. Then no intermediate elements can be inserted into this sequence (to do so would lengthen it), so each pair (a, a_1), (a_1, a_2), $\ldots$, (a_n, b) is in the covering relation, so again (a, b) is in its reflexive transitive closure. This completes the proof. Note how the finiteness of the poset was crucial here. If we let S be the set of all subsets of $\mathbf{N}$ (the set of natural numbers) under the subset relation, then we cannot recover S from its covering relation, since nothing in the covering relation allows us to relate a finite set to an infinite one; thus for example we could not recover the relationship $\{1, 2\} \subset \mathbf{N}$.

33. It is helpful in this exercise to draw the Hasse diagram.

a) Maximal elements are those that do not divide any other elements of the set. In this case 24 and 45 are the only numbers that meet that requirement.

b) Minimal elements are those that are not divisible by any other elements of the set. In this case 3 and 5 are the only numbers that meet that requirement.

c) A greatest element would be one that all the other elements divide. The only two candidates (maximal elements) are 24 and 45, and since neither divides the other, we conclude that there is no greatest element.

d) A least element would be one that divides all the other elements. The only two candidates (minimal elements) are 3 and 5, and since neither divides the other, we conclude that there is no least element.

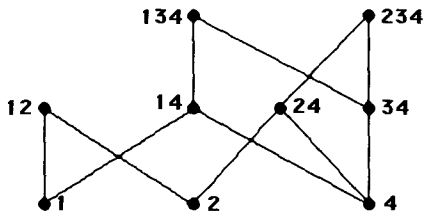
e) We want to find all elements that both 3 and 5 divide. Clearly only 15 and 45 meet this requirement.

f) The least upper bound is 15 since it divides 45 (see part **(e)**).

g) We want to find all elements that divide both 15 and 45. Clearly only 3, 5, and 15 meet this requirement.

h) The number 15 is the greatest lower bound, since both 3 and 5 divide it (see part **(g)**).

35. To help us answer the questions, we will draw the Hasse diagram, with the commas and braces eliminated in the labels, for readability.



a) The maximal elements are the ones without any elements lying above them in the Hasse diagram, namely $\{1,2\}$, $\{1,3,4\}$, and $\{2,3,4\}$.

b) The minimal elements are the ones without any elements lying below them in the Hasse diagram, namely $\{1\}$, $\{2\}$, and $\{4\}$.

c) There is no greatest element, since there is more than one maximal element, none of which is greater than the others.

d) There is no least element, since there is more than one minimal element, none of which is less than the others.

e) The upper bounds are the sets containing both $\{2\}$ and $\{4\}$ as subsets, i.e., the sets containing both 2 and 4 as elements. Pictorially, these are the elements lying above both $\{2\}$ and $\{4\}$ (in the sense of there being a path in the diagram), namely $\{2,4\}$ and $\{2,3,4\}$.

f) The least upper bound is an upper bound that is less than every other upper bound. We found the upper bounds in part **(e)**, and since $\{2,4\}$ is less than (i.e., a subset of) $\{2,3,4\}$, we conclude that $\{2,4\}$ is the least upper bound.

g) To be a lower bound of both $\{1,3,4\}$ and $\{2,3,4\}$, a set must be a subset of each, and so must be a subset of their intersection, $\{3,4\}$. There are only two such subsets in our poset, namely $\{3,4\}$ and $\{4\}$. In the diagram, these are the points which lie below (in the path sense) both $\{1,3,4\}$ and $\{2,3,4\}$.

h) The greatest lower bound is a lower bound that is greater than every other lower bound. We found the lower bounds in part **(g)**, and since $\{3,4\}$ is greater than (i.e., a superset of) $\{4\}$, we conclude that $\{3,4\}$ is the greatest lower bound.

37. First we need to show that lexicographic order is reflexive, i.e., that $(a,b) \preceq (a,b)$; this is true by fiat, since we defined $\preceq$ by adding equality to $\prec$. Next we need to show antisymmetry: if $(a,b) \preceq (c,d)$ and $(a,b) \neq (c,d)$, then $(c,d) \not\preceq (a,b)$. By definition $(a,b) \prec (c,d)$ if and only if either $a \prec c$, or $a = c$ and $b \prec d$. In the first case, by the antisymmetry of the underlying relation, we know that $c \not\prec a$, and similarly in the second case we know that $d \not\prec d$. Thus there is no way that we could have $(c,d) \prec (a,b)$. Finally, for transitivity, let $(a,b) \preceq (c,d) \preceq (e,f)$. We want to show that $(a,b) \preceq (e,f)$. If one of the given inequalities is an equality, then there is nothing to prove, so we may assume that $(a,b) \prec (c,d) \prec (e,f)$. If $a \prec c$, then by the transitivity of the underlying relation, we know that $a \prec e$ and so $(a,b) \prec (e,f)$. Similarly, if $c \prec e$, then again $a \prec e$

and so $(a, b) \prec (e, f)$. The only other way for the given inequalities to hold is if $a = c = e$ and $b \prec d \prec f$. In this case the latter string of inequalities implies that $b \prec f$ and so again by definition $(a, b) \prec (e, f)$.

39. First we must show that $\preceq$ is reflexive. Since $s \preceq_1 s$ and $t \preceq_2 t$ by the reflexivity of these underlying partial orders, $(s, t) \preceq (s, t)$ by definition. For antisymmetry, assume that $(s, t) \preceq (u, v)$ and $(u, v) \preceq (s, t)$. Then by definition $s \preceq_1 u$ and $t \preceq_2 v$, and $u \preceq_1 s$ and $v \preceq_2 t$. By the antisymmetry of the underlying relations, we conclude that $s = u$ and $t = v$, whence $(s, t) = (u, v)$. Finally, for transitivity, suppose that $(s, t) \preceq (u, v) \preceq (w, x)$. This means that $s \preceq_1 u \preceq_1 w$ and $t \preceq_2 v \preceq_2 x$. The transitivity of the underlying partial orders tells us that $s \preceq_1 w$ and $t \preceq_2 x$, whence by definition $(s, t) \preceq (w, x)$.

41. **a)** We argue essentially by contradiction. Suppose that m_1 and m_2 are two maximal elements in a poset that has a greatest element g; we will show that $m_1 = m_2$. Now since g is greatest, we know that $m_1 \preceq g$, and similarly for m_2. But since each m_i is maximal, it cannot be that $m_i \prec g$; hence $m_1 = g = m_2$.

b) The proof is exactly dual to the proof in part **(a)**, so we just copy over that proof, making the appropriate changes in wording. To wit: we argue essentially by contradiction. Suppose that m_1 and m_2 are two minimal elements in a poset that has a least element l; we will show that $m_1 = m_2$. Now since l is least, we know that $l \preceq m_1$, and similarly for m_2. But since each m_i is minimal, it cannot be that $l \prec m_i$; hence $m_1 = l = m_2$.

43. In each case, we need to check whether every pair of elements has both a least upper bound and a greatest lower bound.

a) This is a lattice. If we want to find the l.u.b. or g.l.b. of two elements in the same vertical column of the Hasse diagram, then we simply take the higher or lower (respectively) element. If the elements are in different columns, then to find the g.l.b. we follow the diagonal line upward from the element on the left, and then continue upward on the right, if necessary to reach the element on the right. For example, the l.u.b. of d and c is f; and the l.u.b. of a and e is e. Finding greatest lower bounds in this poset is similar.

b) This is not a lattice. Elements b and c have f, g, and h as upper bounds, but none of them is a l.u.b.

c) This is a lattice. By considering all the pairs of elements, we can verify that every pair of them has a l.u.b. and a g.l.b. For example, b and e have g and a filling these roles, respectively.

45. As usual when trying to extend a theorem from two items to an arbitrary finite number, we will use mathematical induction. The statement we wish to prove is that if S is a subset consisting of n elements from a lattice, where n is a positive integer, then S has a least upper bound and a greatest lower bound. The two proofs are duals of each other, so we will just give the proof for least upper bound here. The basis is $n = 1$, in which case there is really nothing to prove. If $S = \{x\}$, then clearly x is the least upper bound of S. The case $n = 2$ could be singled out for special mention also, since the l.u.b. in that case is guaranteed by the definition of lattice. But there is no need to do so. Instead, we simply assume the inductive hypothesis, that every subset containing n elements has a l.u.b., and prove that every subset S containing $n+1$ elements also has a l.u.b. Pick an arbitrary element $x \in S$, and let $S' = S - \{x\}$. Since S' has only n elements, it has a l.u.b. y, by the inductive hypothesis. Since we are in a lattice, there is an element z that is the l.u.b. of x and y. We will show that in fact z is the least upper bound of S. To do this, we need to show two things: that z is an upper bound, and that every upper bound is greater than or equal to z. For the first statement, let w be an arbitrary element of S; we must show that $w \preceq z$. There are two cases. If $w = x$, then $w \preceq z$ since z is the l.u.b. of x and y. Otherwise, $w \in S'$, and so $w \preceq y$ because y is the l.u.b. of S'. But since z is the l.u.b. of x and y, we also have $y \preceq z$. By transitivity, then, $w \preceq z$. For the second statement, suppose that u is any other upper bound of S; we must show that $z \preceq u$. Since u is an upper bound of S, it is also an upper bound of x and y. But since z is the *least* upper bound of x and y, we know that $z \preceq u$.

47. The needed definitions are in Example 25.

a) No. The authority level of the first pair (1) is less than or equal to (less than, in this case) that of the second (2); but the subset of the first pair is not a subset of that of the second.

b) Yes. The authority level of the first pair (2) is less than or equal to (less than, in this case) that of the second (3); and the subset of the first pair is a subset of that of the second.

c) The classes into which information can flow are those classes whose authority level is at least as high as *Proprietary*, and whose subset is a superset of $\{Cheetah, Puma\}$. We can list these classes: $(Proprietary, \{Cheetah, Puma\})$, $(Restricted, \{Cheetah, Puma\})$, $(Registered, \{Cheetah, Puma\})$, $(Proprietary, \{Cheetah, Puma, Impala\})$, $(Restricted, \{Cheetah, Puma, Impala\})$, and $(Registered, \{Cheetah, Puma, Impala\})$.

d) The classes from which information can flow are those classes whose authority level is at least as low as *Restricted*, and whose subset is a subset of $\{Impala, Puma\}$, namely $(Nonproprietary, \{Impala, Puma\})$, $(Proprietary, \{Impala, Puma\})$, $(Restricted, \{Impala, Puma\})$, $(Nonproprietary, \{Impala\})$, $(Proprietary, \{Impala\})$, $(Restricted, \{Impala\})$, $(Nonproprietary, \{Puma\})$, $(Proprietary, \{Puma\})$, $(Restricted, \{Puma\})$, $(Nonproprietary, \emptyset)$, $(Proprietary, \emptyset)$, and $(Restricted, \emptyset)$.

49. Let Π be the set of all partitions of a set S, with a relation $\preceq$ defined on Π according to the referenced preamble: a partition P_1 is a refinement of P_2 if every set in P_1 is a subset of one of the sets in P_2. We need to verify all the properties of a lattice. First we need to show that $(\Pi, \preceq)$ is a poset, that is, that $\preceq$ is reflexive, antisymmetric, and transitive. For reflexivity, we need to show that $P \preceq P$ for every partition P. This means that every set in P is a subset of one of the sets in P, and this is trivially true, since every set is a subset of itself. For antisymmetry, suppose that $P_1 \preceq P_2$ and $P_2 \preceq P_1$. We must show that $P_1 = P_2$. By the equivalent roles played here by P_1 and P_2, it is enough to show that every $T \in P_1$ (where $T \subseteq S$) is also an element of P_2. Suppose we have such a T. Then since $P_1 \preceq P_2$, there is a set $T' \in P_2$ such that $T \subseteq T'$. But then since $P_2 \preceq P_1$, there is a set $T'' \in P_1$ such that $T' \subseteq T''$. Putting these together, we have $T \subseteq T''$. But P_1 is a partition, and so the elements of P_1 are nonempty and pairwise disjoint. The only way for this to happen if one is a subset of the other is for the two subsets T and T'' to be the same. But this implies that T' (which is caught in the middle) is also equal to T. Thus $T \in P_2$, which is what we were trying to show. Finally, for transitivity, suppose that $P_1 \preceq P_2$ and $P_2 \preceq P_3$. We must show that $P_1 \preceq P_3$. To this end, we take an arbitrary element $T \in P_1$. Then there is a set $T' \in P_2$ such that $T \subseteq T'$. But then since $P_2 \preceq P_3$, there is a set $T'' \in P_3$ such that $T' \subseteq T''$. Putting these together, we have $T \subseteq T''$. This demonstrates that $P_1 \preceq P_3$.

Next we have to show that every two partitions P_1 and P_2 have a least upper bound and a greatest lower bound in Π. We will show that their greatest lower bound is their "coarsest common refinement", namely the partition P whose subsets are all the nonempty sets of the form $T_1 \cap T_2$, where $T_1 \in P_1$ and $T_2 \in P_2$. As an example, if $P_1 = \{\{1,2,3\},\{4\},\{5\}\}$ and $P_2 = \{\{1,2\},\{3,4\},\{5\}\}$, then the coarsest common refinement is $P = \{\{1,2\},\{3\},\{4\},\{5\}\}$. First, we need to check that this is a partition. It certainly is a set of nonempty subsets of S. It is pairwise disjoint, because the only way an element could be in $T_1 \cap T_2 \cap T_1' \cap T_2'$ if $T_1 \cap T_2 \neq T_1' \cap T_2'$ is for that element to be in both $T_1 \cap T_1'$ and $T_2 \cap T_2'$, which means that $T_1 = T_1'$ and $T_2 = T_2'$, a contradiction. And it covers all of S, because if $x \in S$, then $x \in T_1$ for some $T_1 \in P_1$, and $x \in T_2$ for some $T_2 \in P_2$, and so $x \in T_1 \cap T_2 \in P$. Second, we need to check that P is a refinement of both P_1 and P_2. So suppose $T \in P$. Then $T = T_1 \cap T_2$, for some $T_1 \in P_1$ and $T_2 \in P_2$. It follows that $T \subseteq T_1$ and $T \subseteq T_2$. But then T_1 and T_2 satisfy the requirements in the definition of refinement. Third, we need to check that if P' is any other common refinement of both P_1 and P_2, then P' is also a refinement of P. To this end, suppose that $T \in P'$. Then by definition of refinement, there are subsets $T_1 \in P_1$ and $T_2 \in P_2$ such that $T \subseteq T_1$ and $T \subseteq T_2$. Therefore $T \subseteq T_1 \cap T_2$. But $T_1 \cap T_2 \in P$, and our proof for greatest lower bounds is complete.

It's a little harder to state the definition of the least upper bound (which again we'll call P) of two given partitions P_1 and P_2. Essentially it is just the set of all minimal nonempty subsets of S that do not "split

apart" any element of either P_1 or P_2. (In the example above, it is $\{\{1,2,3,4\},\{5\}\}$.) It will be a little easier if we define it in terms of an equivalence relation rather than a partition. Note that from this point of view, one equivalence relation is a refinement of a second equivalence relation if whenever two elements are related by the first relation, then they are related by the second. The equivalence relation determining P is the relation in which $x \in S$ is related to $y \in S$ if there is a "path" (a sequence) $x = x_0, x_1, x_2, \ldots, x_n = y$, for some $n \geq 0$, such that for each i from 1 to n, x_{i-1} and x_i are in the same element of partition P_1 or of partition P_2 (in other words, x_{i-1} and x_i are related either by the equivalence relation corresponding to P_1 or by that corresponding to P_2). It is clear that this is an equivalence relation: it is reflexive by taking $n = 0$; it is symmetric by following the path backwards; and it is transitive by composing paths. It is also clear that P_1 (and P_2 similarly) is a refinement of this partition, since if two elements of S are in the same equivalence class in P_1, then we can take $n = 1$ in our path definition to see that they are in the same equivalence class in P. Thus P is an upper bound of both P_1 and P_2. Finally, we must show that P is the *least* upper bound, that is, a refinement of every other upper bound. This is clear from our construction: we only forced two elements of S to be related (i.e., in the same class of the partition) when they *had* to be related in order to enable P_1 and P_2 to be refinements. Therefore if two elements are related by P, then they have to be related by every equivalence relation (partition) Q of which both P_1 and P_2 are refinements; so P is a refinement of Q.

51. This follows immediately from Exercise 45. To be more specific, according to Exercise 45, there is a least upper bound (respectively, a greatest lower bound) for the entire finite lattice. This element is by definition a greatest element (respectively, a least element).

53. We need to show that every nonempty subset of $\mathbf{Z}^+ \times \mathbf{Z}^+$ has a least element under lexicographic order. Given such a subset S, look at the set S_1 of positive integers that occur as first coordinates in elements of S. Let m_1 be the least element of S_1, which exists since $\mathbf{Z}^+$ is well-ordered under $\leq$. Let S' be the subset of S consisting of those pairs that have m_1 as their first coordinate. Thus S' is clearly nonempty, and by the definition of lexicographic order, every element of S' is less than every element in $S - S'$. Now let S_2 be the set of positive integers that occur as second coordinates in elements of S', and let m_2 be the least element of S_2. Then clearly the element (m_1, m_2) is the least element of S' and hence is the least element of S.

55. If x is an integer in a decreasing sequence of elements of this poset, then at most $|x|$ elements can follow x in the sequence, namely integers whose absolute values are $|x| - 1, |x| - 2, \ldots, 1, 0$. Therefore there can be no infinite decreasing sequence. This is not a totally ordered set, since 5 and -5, for example, are incomparable; from the definition given here, it is neither true that $5 \prec -5$ nor that $-5 \prec 5$, because neither one of $|5|$ or $|-5|$ is less than the other (they are equal).

57. We know from elementary arithmetic that $\mathbf{Q}$ is totally ordered by $<$, and so perforce it is a partially ordered set. To be precise, to find which of two rational numbers is larger, write them with a positive common denominator and compare numerators. To show that this set is dense, suppose $x < y$ are two rational numbers. Let z be their average, i.e., $(x + y)/2$. Since the set of rational numbers is closed under addition and division, z is also a rational number, and it is easy to show that $x < z < y$.

59. Let $(S, \preceq)$ be a partially ordered set. From the definitions of well-ordered, totally ordered, and well-founded, it is clear that what we have to show is that every nonempty subset of S contains a least element if and only if there is no infinite decreasing sequence of elements $a_1, a_2, a_3, \ldots$ in S (i.e., where $a_{i+1} \prec a_i$ for all i). One direction is clear: An infinite decreasing sequence of elements has no least element. Conversely, let A be any nonempty subset of S that has no least element. Since A is nonempty, let a_1 be any element of A. Since a_1 is not the least element of A, there is some $a_2 \in A$ smaller than a_1, i.e., $a_2 \prec a_1$. Since a_2 is not the least

element of A, A must contain an element a_3 with $a_3 \prec a_2$. We continue in this manner, giving us an infinite decreasing sequence in S. Note that this proof is nonconstructive; it uses what set theorists call the Axiom of Choice.

61. We need to peel elements off the bottom of the Hasse diagram. We can begin with a, b, or c. Suppose we decide to start with a. Next we may choose any minimal element of what remains after we have removed a; only b and c meet this requirement. Suppose we choose b next. Then c, d, and e are minimal elements in what remains, so any of those can come next. We continue in this manner until we have listed and removed all the elements. One possible order, then, is $a \prec_t b \prec_t d \prec_t e \prec_t c \prec_t f \prec_t g \prec_t h \prec_t i \prec_t j \prec_t k \prec_t m \prec_t l$.

63. Clearly 1 must come first, and 20 must follow each element except possibly 12. The relative positions of 2, 4, and 12 are fixed. The 5 can go anywhere, as long as it lies between 1 and 20. Following these guidelines, we see that the following seven total orderings are the ones compatible with the given relation: $1 \prec 5 \prec 2 \prec 4 \prec 12 \prec 20$, $1 \prec 2 \prec 5 \prec 4 \prec 12 \prec 20$, $1 \prec 2 \prec 4 \prec 5 \prec 12 \prec 20$, $1 \prec 2 \prec 4 \prec 12 \prec 5 \prec 20$, $1 \prec 5 \prec 2 \prec 4 \prec 20 \prec 12$, $1 \prec 2 \prec 5 \prec 4 \prec 20 \prec 12$, $1 \prec 2 \prec 4 \prec 5 \prec 20 \prec 12$.

65. There are a few restrictions, but there are many choices, so we will get many (18) compatible total orderings. Note that A and C must precede B; B and E must precede F; B must precede D; and G must come last. We can therefore make the following list: $A \prec C \prec E \prec B \prec D \prec F \prec G$, $A \prec E \prec C \prec B \prec D \prec F \prec G$, $C \prec A \prec E \prec B \prec D \prec F \prec G$, $C \prec E \prec A \prec B \prec D \prec F \prec G$, $E \prec A \prec C \prec B \prec D \prec F \prec G$, $E \prec C \prec A \prec B \prec D \prec F \prec G$, $A \prec C \prec B \prec E \prec D \prec F \prec G$, $C \prec A \prec B \prec E \prec D \prec F \prec G$, $A \prec C \prec B \prec D \prec E \prec F \prec G$, $C \prec A \prec B \prec D \prec E \prec F \prec G$, $A \prec C \prec E \prec B \prec F \prec D \prec G$, $A \prec E \prec C \prec B \prec F \prec D \prec G$, $C \prec A \prec E \prec B \prec F \prec D \prec G$, $C \prec E \prec A \prec B \prec F \prec D \prec G$, $E \prec A \prec C \prec B \prec F \prec D \prec G$, $E \prec C \prec A \prec B \prec F \prec D \prec G$, $A \prec C \prec B \prec E \prec F \prec D \prec G$, and $C \prec A \prec B \prec E \prec F \prec D \prec G$.

67. We need to find a total order compatible with this partial order. We work from the bottom up, writing down a task (vertex in the diagram) and removing it from the diagram, so that at each stage we choose a vertex with no vertices below it. One such order is: Determine user needs $\prec$ Write functional requirements $\prec$ Set up test sites $\prec$ Develop system requirements $\prec$ Develop module A $\prec$ Develop module C $\prec$ Develop module B $\prec$ Write documentation $\prec$ Integrate modules $\prec$ α test $\prec$ β test $\prec$ Completion.

GUIDE TO REVIEW QUESTIONS FOR CHAPTER 9

1. **a)** See p. 575 (which refers to the definition on p. 573). **b)** See Example 6 in Section 9.1.

2. **a)** See p. 576. **b)** See p. 577. **c)** See p. 577. **d)** See p. 578.

3. **a)** $\{(1,1),(1,2),(2,1),(2,2),(2,3),(3,2),(3,3),(4,4)\}$
b) $\emptyset$ **c)** $\{(1,1),(1,2),(2,2),(2,3),(3,3),(4,4)\}$
d) $\{(1,1),(2,2),(3,3),(4,4)\}$ **e)** $\{(1,1),(2,2),(3,3),(4,4)\}$

4. **a)** See Example 16 in Section 9.1. **b)** See Exercise 47a in Section 9.1.
c) See Exercise 47b in Section 9.1.

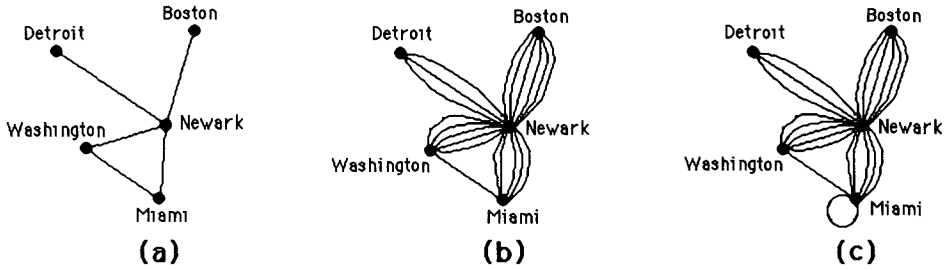
5. **a)** See pp. 584–585. **b)** Take the projection $P_{1,4,5}$.
c) First rearrange the order of the fields in the relations, so that the first is in the order address, telephone number, name, major, and the second is in the order name, major, student number, number of credit hours. Then form the join J_2, to get a single relation with the fields in the order address, telephone number, name, major, student number, number of credit hours. Finally, if desired, rearrange the fields to a more natural order.

CHAPTER 10
Graphs

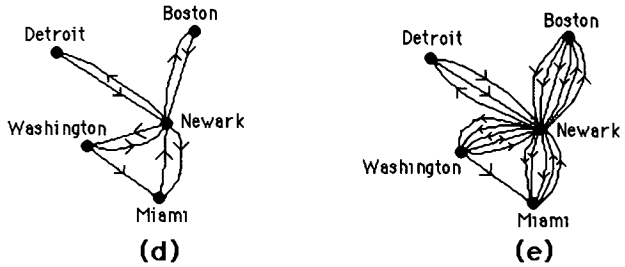
SECTION 10.1 Graphs and Graph Models

The examples and exercises give a good picture of the ways in which graphs can model various real world applications. In constructing graph models you need to determine what the vertices will represent, what the edges will represent, whether the edges will be directed or undirected, whether loops should be allowed, and whether a simple graph or multigraph is more appropriate.

1. In part **(a)** we have a simple graph, with undirected edges, no loops or multiple edges. In part **(b)** we have a multigraph, since there are multiple edges (making the figure somewhat less than ideal visually). In part **(c)** we have the same picture as in part **(b)** except that there is now a loop at one vertex; thus this is a pseudograph.



In part **(d)** we have a directed graph, the directions of the edges telling the directions of the flights; note that the **antiparallel edges** (pairs of the form (u, v) and (v, u)) are not parallel. In part **(e)** we have a directed multigraph, since there are parallel edges.



3. This is a simple graph; the edges are undirected, and there are no parallel edges or loops.

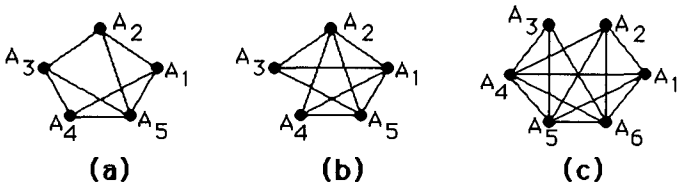
5. This is a pseudograph; the edges are undirected, but there are loops and parallel edges.

7. This is a directed graph; the edges are directed, but there are no parallel edges. (Loops and antiparallel edges—see the solution to Exercise 1d for a definition—are allowed in a directed graph.)

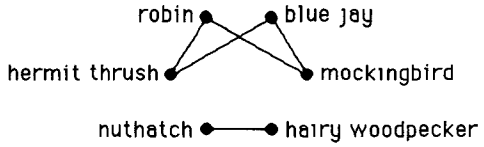
9. This is a directed multigraph; the edges are directed, and there is a set of parallel edges.

11. In a simple graph, edges are undirected. To show that R is symmetric we must show that if uRv, then vRu. If uRv, then there is an edge associated with $\{u, v\}$. But $\{u, v\} = \{v, u\}$, so this edge is associated with $\{v, u\}$ and therefore vRu. A simple graph does not allow loops; that is if there is an edge associated with $\{u, v\}$, then $u \neq v$. Thus uRu never holds, and so by definition R is irreflexive.

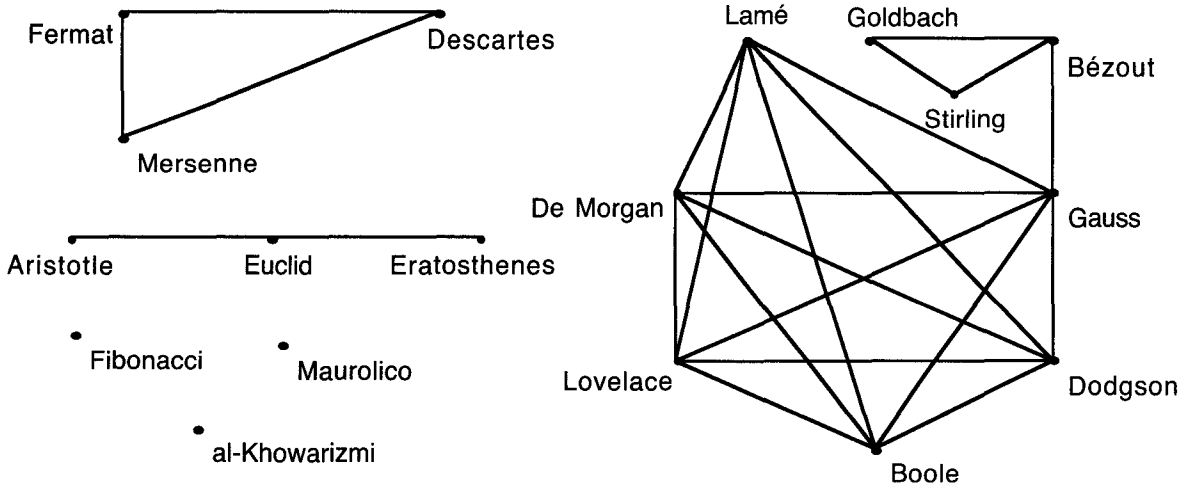
13. In each case we draw a picture of the graph in question. All are simple graphs. An edge is drawn between two vertices if the sets for the two vertices have at least one element in common. For example, in part **(a)** there is an edge between vertices A_1 and A_2 because there is at least one element common to A_1 and A_2 (in fact there are three such elements). There is no edge between A_1 and A_3 since $A_1 \cap A_3 = \emptyset$.



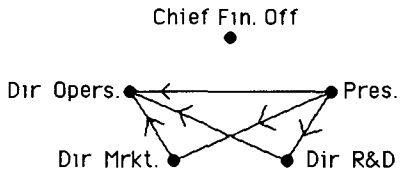
15. We draw a picture of the graph in question, which is a simple graph. Two vertices are joined by an edge if we are told that the species compete (such as robin and mockingbird) but there is no edge between pairs of species that are not given as competitors (such as robin and blue jay).



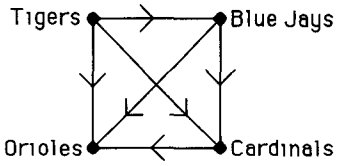
17. Here are the persons to be included, listed in order of birth year: Aristotle (384–322 B.C.E.), Euclid (325–265 B.C.E.), Eratosthenes (276–194 B.C.E.), al-Khowarizmi (780–850), Fibonacci (1170–1250), Maurolico (1494–1575), Mersenne (1588–1648), Descartes (1596–1650), Fermat (1601–1665), Goldbach (1690–1764), Stirling (1692–1770), Bézout (1730–1783), Gauss (1777–1855), Lamé (1795–1870), De Morgan (1806–1871), Lovelace (1815–1852), Boole (1815–1864), and Dodgson (1832–1898). We draw the graph by connecting two people if their date ranges overlap. Note that there is a complete subgraph (see Section 10.2) consisting of the last six people listed. A few of the vertices are isolated (again see Section 10.2). In all our graph has 17 vertices and 22 edges. A graph like this is called an **interval graph**, since each vertex can be associated with an interval of real numbers; it is a special case of an **intersection graph**, where two vertices are adjacent if the sets associated with those vertices have a nonempty intersection (see Exercise 13).



19. We draw a picture of the graph in question, which is a directed graph. We draw an edge from u to v if we are told that u can influence v. For instance the Chief Financial Officer is an isolated vertex since she is influenced by no one and influences no one.



21. We draw a picture of the graph in question, which is a directed graph. We draw an edge from u to v if we are told that u beat v.



23. We could compile a list of phone numbers (the labels on the vertices) in the February call graph that were not present in January, and a list of the January numbers missing in February. For each number in each list, we could make a list of the numbers they called or were called by, using the edges in the call graphs. Then we could look for February lists that were very similar to January lists. If we found a new February number that had almost the same calling pattern as a defunct January number, then we might suspect that these numbers belonged to the same person, who had recently changed his or her number.

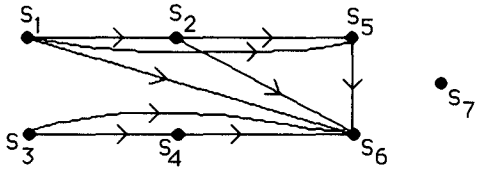
25. For each e-mail address (the labels on the vertices), we could make a list of the other addresses they sent messages to or received messages from. If we see two addresses that had almost the same communication pattern, then we might suspect that these addresses belonged to the same person, who had recently changed his or her e-mail address.

27. The vertices represent the people at the party. Because it is possible that a knows b's name but not vice versa, we need a directed graph. We will include an edge associated with (u, v) if and only if u knows v's name. There is no need for multiple edges (either a knows b's name or he doesn't). One could argue that we should not clutter the model with loops, because obviously everyone knows her own name. On the other hand, it certainly would not be wrong to include loops, especially if we took the instructions literally.

29. We should use a directed graph, with the vertices being the courses and the edges showing the prerequisite relationship. Specifically, an edge from u to v means that course u is a prerequisite for course v. Courses that do not have any prerequisites are the courses with in-degree 0, and courses that are not the prerequisite for any other courses have out-degree 0. An interesting question would be how to model courses that are co-requisites (in two different senses—either courses u and v must be taken at the same time, or course u must be taken before course v or in the same semester as course v).

31. For this to be interesting, we want the graph to model all marriages, not just ones that are currently active. (In the latter case, for the Western world, there would be at most one edge incident to each vertex.) So we let the set of vertices be a set of people (for example, all the people in North America who lived at any point in the 20th century), and two vertices are joined by an edge if the two people were ever married. Since laws in the 20th century allowed only marriages between persons of the opposite sex, and ignoring complications caused by sex-change operations, we note that this graph has the property that there are two types of vertices (men and women), and every edge joins vertices of opposite types. In the next section we learn that the word used to describe a graph like this is *bipartite*.

33. We draw a picture of the directed graph in question. There is an edge from u to v if the assignment made in u can possibly influence the assignment made in v. For example, there is an edge from S_3 to S_6, since the assignment in S_3 changes the value of y, which then influences the value of z (in S_4) and hence has a bearing on S_6. We assume that the statements are to be executed in the given order, so, for example, we do not draw an edge from S_5 to S_2.



35. The vertices in the directed graph represent people in the group. We put a directed edge into our directed graph from every vertex A to every vertex $B \neq A$ (we do not need loops), and furthermore we label that edge with one of the three labels L, D, or N. Let us see how to incorporate this into the mathematical definition. Let us call such a thing a directed graph with labeled edges. It is defined to be a triple (V, E, f), where (V, E) is a directed graph (i.e., V is a set of vertices and E is a set of ordered pairs of elements of V) and f is a function from E to the set $\{L, D, N\}$. Here we are simply thinking of $f(e)$ as the attitude of the person at the tail (initial vertex—see Section 10.2) of e toward the person at the head (terminal vertex) of e.

SECTION 10.2 Graph Terminology and Special Types of Graphs

Graph theory is sometimes jokingly called the "theory of definitions," because so many terms can be—and have been—defined for graphs. A few of the most important concepts are given in this section; others appear in the rest of this chapter and the next, in the exposition and in the exercises. As usual with definitions, it is important to understand exactly what they are saying. You should construct some examples for each definition you encounter—examples both of the thing being defined and of its absence. Some students find it useful to build a dictionary as they read, including their examples along with the formal definitions.

The handshaking theorem (that the sum of the degrees of the vertices in a graph equals twice the number of edges), although trivial to prove, is quite handy, as Exercise 55, for example, illustrates. Be sure to look at Exercise 43, which deals with the problem of when a sequence of numbers can possibly be the degrees of the vertices of a simple graph. Some interesting subtleties arise there, as you will discover when you try to draw the graphs. Many arguments in graph theory tend to be rather ad hoc, really getting down to the nitty gritty, and Exercise 43c is a good example. Exercise 51 is really a combinatorial problem; such problems abound in graph theory, and entire books have been written on counting graphs of various types. The notion of **complementary graph**, *introduced in Exercise 59, will appear again later in this chapter, so it would be wise to look at the exercises dealing with it.*

1. There are 6 vertices here, and 6 edges. The degree of each vertex is the number of edges incident to it. Thus $\deg(a) = 2$, $\deg(b) = 4$, $\deg(c) = 1$ (and hence c is pendant), $\deg(d) = 0$ (and hence d is isolated), $\deg(e) = 2$, and $\deg(f) = 3$. Note that the sum of the degrees is $2 + 4 + 1 + 0 + 2 + 3 = 12$, which is twice the number of edges.

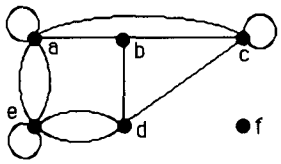
3. There are 9 vertices here, and 12 edges. The degree of each vertex is the number of edges incident to it. Thus $\deg(a) = 3$, $\deg(b) = 2$, $\deg(c) = 4$, $\deg(d) = 0$ (and hence d is isolated), $\deg(e) = 6$, $\deg(f) = 0$ (and hence f is isolated), $\deg(g) = 4$, $\deg(h) = 2$, and $\deg(i) = 3$. Note that the sum of the degrees is $3 + 2 + 4 + 0 + 6 + 0 + 4 + 2 + 3 = 24$, which is twice the number of edges.

5. By Theorem 2 the number of vertices of odd degree must be even. Hence there cannot be a graph with 15 vertices of odd degree 5. (We assume that the problem was meant to imply that the graph contained only these 15 vertices.)

7. This directed graph has 4 vertices and 7 edges. The in-degree of vertex a is $\deg^-(a) = 3$ since there are 3 edges with a as their terminal vertex; its out-degree is $\deg^+(a) = 1$ since only the loop has a as its initial vertex. Similarly we have $\deg^-(b) = 1$, $\deg^+(b) = 2$, $\deg^-(c) = 2$, $\deg^+(c) = 1$, $\deg^-(d) = 1$, and $\deg^+(d) = 3$. As a check we see that the sum of the in-degrees and the sum of the out-degrees are equal (both are equal to 7).

9. This directed multigraph has 5 vertices and 13 edges. The in-degree of vertex a is $\deg^-(a) = 6$ since there are 6 edges with a as their terminal vertex; its out-degree is $\deg^+(a) = 1$. Similarly we have $\deg^-(b) = 1$, $\deg^+(b) = 5$, $\deg^-(c) = 2$, $\deg^+(c) = 5$, $\deg^-(d) = 4$, $\deg^+(d) = 2$, $\deg^-(e) = 0$, and $\deg^+(e) = 0$ (vertex e is isolated). As a check we see that the sum of the in-degrees and the sum of the out-degrees are both equal to the number of edges (13).

11. To form the underlying undirected graph we simply take all the arrows off the edges. Thus, for example, the edges from e to d and from d to e become a pair of parallel edges between e and d.



13. Since a person is joined by an edge to each of his or her collaborators, the degree of v is the number of collaborators v has. Similarly, the neighborhood of a vertex is the set of coauthors of the person represented by that vertex. An isolated vertex represents a person who has no coauthors (he or she has published only single-authored papers), and a pendant vertex represents a person who has published with just one other person.

15. Since there is a directed edge from u to v for each call made by u to v, the in-degree of v is the number of calls v received, and the out-degree of u is the number of calls u made. The degree of a vertex in the undirected version is just the sum of these, which is therefore the number of calls the vertex was involved in.

17. Since there is a directed edge from u to v to represent the event that u beat v when they played, the in-degree of v must be the number of teams that beat v, and the out-degree of u must be the number of teams that u beat. In other words, the pair $(\deg^+(v), \deg^-(v))$ is the win-loss record of v.

19. Model the friendship relation with a simple undirected graph in which the vertices are people in the group, and two vertices are adjacent if those two people are friends. The degree of a vertex is the number of friends in the group that person has. By Exercise 18, there are two vertices with the same degree, which means that there are two people in the group with the same number of friends in the group.

21. To show that this graph is bipartite we can exhibit the parts and note that indeed every edge joins vertices in different parts. Take $\{e\}$ to be one part and $\{a, b, c, d\}$ to be the other (in fact there is no choice in the matter). Each edge joins a vertex in one part to a vertex in the other. This graph is the complete bipartite graph $K_{1,4}$.

23. To show that a graph is not bipartite we must give a proof that there is no possible way to specify the parts. (There is another good way to characterize nonbipartite graphs, but it takes some notions not introduced until Section 10.4.) We can show that this graph is not bipartite by the pigeonhole principle. Consider the vertices b, c, and f. They form a triangle—each is joined by an edge to the other two. By the pigeonhole principle, at least two of them must be in the same part of any proposed bipartition. Therefore there would be an edge joining two vertices in the same part, a contradiction to the definition of a bipartite graph. Thus this graph is not bipartite.

An alternative way to look at this is given by Theorem 4. Because of the triangle, it is impossible to color the vertices to satisfy the condition given there.

25. As in Exercise 23, we can show that this graph is not bipartite by looking at a triangle, in this case the triangle formed by vertices b, d, and e. Each of these vertices is joined by an edge to the other two. By the pigeonhole principle, at least two of them must be in the same part of any proposed bipartition. Therefore there would be an edge joining two vertices in the same part, a contradiction to the definition of a bipartite graph. Thus this graph is not bipartite.

27. **a)** The bipartite graph has vertices h, s, n, and w representing the support areas and P, Q, R, and S representing the employees. The qualifications are modeled by the bipartite graph with edges Ph, Pn, Pw, Qs, Qn, Rn, Rw, Sh, and Ss.

b) Since every vertex representing an area has degree at least 2, the condition in Hall's theorem is satisfied for sets of size less than 3. We can easily check that the number of employees qualified for each of the four subsets of size 3 is at least 3, and clearly the number of employees qualified for each of the subsets of size 4 has size 4.

c) The answer is not unique; one complete matching is $\{Pn, Qs, Rw, Sh\}$, which is easily found by inspection.

29. The partite sets are the set of women ($\{$Tina, Uma, Vandana, Xia, Zelda$\}$) and the set of men ($\{$Anil, Barry, Emilio, Sandeep, Teja$\}$). We will use first letters for convenience (but J for Teja). The given information tells us that we have edges AV, AZ, BT, BX, BU, ET, EZ, JT, JZ, ST, and SV in our graph. We do not put an edge between a man and a woman he is not willing to marry. By inspection we find that the condition in Hall's theorem is violated by $\{U, X\}$, because these two vertices are adjacent only to B. In other words, only Barry is willing to marry Uma and Xia, so there can be no matching.

31. We model this with an undirected bipartite graph, with the men and the women represented by the vertices in the two parts and an edge between two vertices if they are willing to marry each other. By Hall's theorem, it is enough to show that for every set S of women, the set $N(S)$ of men willing to marry them has cardinality at least $|S|$. A clever way to prove this is by counting edges. Let m be the number of edges between S and $N(S)$. Since every vertex in S has degree k, it follows that $m = k|S|$. Because these edges are incident to $N(S)$, it follows that $m \leq k|N(S)|$. Combining these two facts gives $k|S| \leq k|N(S)|$, so $|N(S)| \geq |S|$, as desired.

33. **a)** By definition, the vertices are a, b, c, and f, and the edges are all the edges of the given graph joining vertices in this list, namely ab, af, bc, and bf.

b) Contracting edge bf merges the vertices b and f into a new vertex; call it x. Edges ab and af are replaced by edge ax; edges eb and ef are replaced by edge ex; and edge cb is replaced by edge cx. Vertex d continues to be an isolated vertex in the contracted graph.

35. **a)** Obviously K_n has n vertices. It has $C(n, 2) = n(n-1)/2$ edges, since each unordered pair of distinct vertices is an edge.

b) Obviously C_n has n vertices. Just as obviously it has n edges.

c) The wheel W_n is the same as C_n with an extra vertex and n extra edges incident to that vertex. Therefore it has $n+1$ vertices and $n+n=2n$ edges.

d) By definition $K_{m,n}$ has $m+n$ vertices. Since it has one edge for each choice of a vertex in the one part and a vertex in the other part, it has mn edges.

e) Since the vertices of Q_n are the bit strings of length n, there are 2^n vertices. Each vertex has degree n, since there are n strings that differ from any given string in exactly one bit (any one of the n different bits can be changed). Thus the sum of the degrees is $n2^n$. Since this must equal twice the number of edges (by the handshaking theorem), we know that there are $n2^n/2 = n2^{n-1}$ edges.

37. In each case we just record the degrees of the vertices in a list, from largest to smallest.

a) Each of the four vertices is adjacent to each of the other three vertices, so the degree sequence is $3,3,3,3$.

b) Each of the four vertices is adjacent to its two neighbors in the cycle, so the degree sequence is $2,2,2,2$.

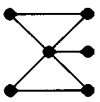
c) Each of the four vertices on the rim of the wheel is adjacent to each of its two neighbors on the rim, as well as to the middle vertex. The middle vertex is adjacent to the four rim vertices. Therefore the degree sequence is $4,3,3,3,3$.

d) Each of the vertices in the part of size two is adjacent to each of the three vertices in the part of size three, and vice versa, so the degree sequence is $3,3,2,2,2$.

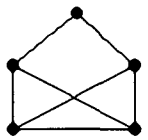
e) Each of the eight vertices in the cube is adjacent to three others (for example, 000 is adjacent to 001, 010, and 100. Therefore the degree sequence is $3,3,3,3,3,3,3,3$.

39. Each of the n vertices is adjacent to each of the other $n-1$ vertices, so the degree sequence is simply $n-1, n-1, \ldots, n-1$, with n terms in the sequence.

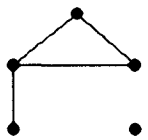
41. The number of edges is half the sum of the degrees (Theorem 1). Therefore this graph has $(5+2+2+2+2+1)/2 = 7$ edges. A picture of this graph is shown here (it is essentially unique).



43. There is no such graph in part **(b)**, since the sum of the degrees is odd (and also because a simple graph with 5 vertices cannot have any degrees greater than 4). Similarly, the odd degree sum prohibits the existence of graphs with the degree sequences given in part **(d)** and part **(f)**. There is no such graph in part **(c)**, since the existence of two vertices of degree 4 implies that there are two vertices each joined by an edge to every other vertex. This means that the degree of each vertex has to be at least 2, and there can be no vertex of degree 1. The graphs for part **(a)** and part **(e)** are shown below; one can draw them after just a little trial and error.



(a)



(e)

45. We need to prove two conditional statements. First, suppose that $d_1, d_2, \ldots, d_n$ is graphic. We must show that the sequence whose terms are $d_2-1, d_3-1, \ldots, d_{d_1+1}-1, d_{d_1+2}, d_{d_1+3}, \ldots, d_n$ is graphic once it is put into nonincreasing order. Apparently what we want to do is to remove the vertex of highest degree (d_1) from a graph with the original degree sequence and reduce by 1 the degrees of the vertices to which it is

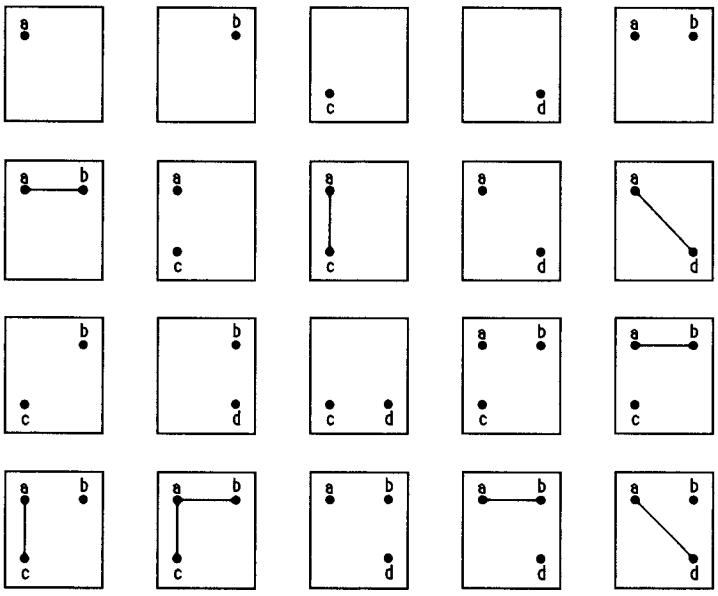
adjacent, but we also want to make sure that those vertices are the ones with the highest degrees among the remaining vertices. In Exercise 44 it is proved that if the original sequence is graphic, then in fact there is a graph having this degree sequence in which the vertex of degree d_1 is adjacent to the vertices of degrees d_2, $d_3, \ldots, d_{d_1+1}$. Thus our plan works, and we have a graph whose degree sequence is as desired.

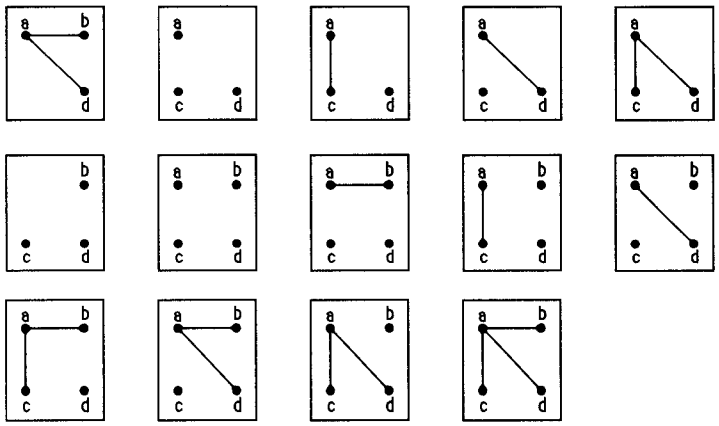
Conversely, suppose that $d_1, d_2, \ldots, d_n$ is a nonincreasing sequence such that the sequence $d_2 - 1$, $d_3 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, d_{d_1+3}, \ldots, d_n$ is graphic once it is put into nonincreasing order. Take a graph with this latter degree sequence, where vertex v_i has degree $d_i - 1$ for $2 \leq i \leq d_1 + 1$ and vertex v_i has degree d_i for $d_1 + 2 \leq i \leq n$. Adjoin one new vertex (call it v_1), and put in an edge from v_1 to each of the vertices $v_2, v_3, \ldots, v_{d_1+1}$. Then clearly the resulting graph has degree sequence $d_1, d_2, \ldots, d_n$.

47. Let $d_1, d_2, \ldots, d_n$ be a nonincreasing sequence of nonnegative integers with an even sum. We want to construct a pseudograph with this as its degree sequence. Even degrees can be achieved using only loops, each of which contributes 2 to the count of its endpoint; vertices of odd degrees will need a non-loop edge, but one will suffice (the rest of the count at that vertex will be made up by loops). Following the hint, we take vertices $v_1, v_2, \ldots, v_n$ and put $\lfloor d_i/2 \rfloor$ loops at vertex v_i, for $i = 1, 2, \ldots, n$. For each i, vertex v_i now has degree either d_i (if d_i is even) or $d_i - 1$ (if d_i is odd). Because the original sum was even, the number of vertices falling into the latter category is even. If there are $2k$ such vertices, pair them up arbitrarily, and put in k more edges, one joining the vertices in each pair. The resulting graph will have degree sequence $d_1, d_2, \ldots, d_n$.

49. We will count the subgraphs in terms of the number of vertices they contain. There are clearly just 3 subgraphs consisting of just one vertex. If a subgraph is to have two vertices, then there are $C(3, 2) = 3$ ways to choose the vertices, and then 2 ways in each case to decide whether or not to include the edge joining them. This gives us $3 \cdot 2 = 6$ subgraphs with two vertices. If a subgraph is to have all three vertices, then there are $2^3 = 8$ ways to decide whether or not to include each of the edges. Thus our answer is $3 + 6 + 8 = 17$.

51. This graph has a lot of subgraphs. First of all, any nonempty subset of the vertex set can be the vertex set for a subgraph, and there are 15 such subsets. If the set of vertices of the subgraph does not contain vertex a, then the subgraph can of course have no edges. If it does contain vertex a, then it can contain or fail to contain each edge from a to whichever other vertices are included. A careful enumeration of all the possibilities gives the 34 graphs shown below.





53. **a)** The complete graph K_n is regular for all values of $n \geq 1$, since the degree of each vertex is $n-1$.

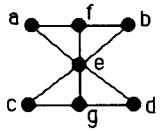
b) The degree of each vertex of C_n is 2 for all n for which C_n is defined, namely $n \geq 3$, so C_n is regular for all these values of n.

c) The degree of the middle vertex of the wheel W_n is n, and the degree of the vertices on the "rim" is 3. Therefore W_n is regular if and only if $n = 3$. Of course W_3 is the same as K_4.

d) The cube Q_n is regular for all values of $n \geq 0$, since the degree of each vertex in Q_n is n. (Note that Q_0 is the graph with 1 vertex.)

55. If a graph is regular of degree 4 and has n vertices, then by the handshaking theorem it has $4n/2 = 2n$ edges. Since we are told that there are 10 edges, we just need to solve $2n = 10$. Thus the graph has 5 vertices. The complete graph K_5 is one such graph (and the only simple one).

57. We draw the answer by superimposing the graphs (keeping the positions of the vertices the same).



59. **a)** The complement of a complete graph is a graph with no edges.

b) Since all the edges between the parts are present in $K_{m,n}$, but none of the edges between vertices in the same part are, the complement must consist precisely of the disjoint union of a K_m and a K_n, i.e., the graph containing all the edges joining two vertices in the same part and no edges joining vertices in different parts.

c) There is really no better way to describe this graph than simply by saying it is the complement of C_n. One representation would be to take as vertex set the integers from 1 to n, inclusive, with an edge between distinct vertices i and j as long as i and j do not differ by ± 1, modulo n.

d) Again, there is really no better way to describe this graph than simply by saying it is the complement of Q_n. One representation would be to take as vertex set the bit strings of length n, with two vertices joined by an edge if the bit strings differ in more than one bit.

61. Since K_v has $C(v,2) = v(v-1)/2$ edges, and since $\overline{G}$ has all the edges of K_v that G is missing, it is clear that $\overline{G}$ has $[v(v-1)/2] - e$ edges.

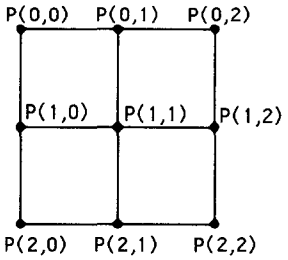
63. If G has n vertices, then the degree of vertex v in $\overline{G}$ is $n-1$ minus the degree of v in G (there will be an edge in $\overline{G}$ from v to each of the $n-1$ other vertices that v is not adjacent to in G). The order of the sequence will reverse, of course, because if $d_i \geq d_j$, then $n-1-d_i \leq n-1-d_j$. Therefore the degree sequence of $\overline{G}$ will be $n-1-d_n$, $n-1-d_{n-1}$, $\ldots$, $n-1-d_2$, $n-1-d_1$.

65. Consider the graph $G \cup \overline{G}$. Its vertex set is clearly the vertex set of G; therefore it has n vertices. If u and v are any two distinct vertices of $G \cup \overline{G}$, then either the edge between u and v is in G, or else by definition it is in $\overline{G}$. Therefore by definition of union, it is in $G \cup \overline{G}$. Thus by definition $G \cup \overline{G}$ is the complete graph K_n.

67. These pictures are identical to the figures in those exercises, with one change, namely that all the arrowheads are turned around. For example, rather than there being a directed edge from a to b in #7, there is an edge from b to a. Note that the loops are unaffected by changing the direction of the arrowhead—a loop from a vertex to itself is the same, whether the drawing of it shows the direction to be clockwise or counterclockwise.

69. It is clear from the definition of converse that a directed graph $G = (V, E)$ is its own converse if and only if it satisfies the condition that $(u, v) \in E$ if and only if $(u, v) \in E$. But this is precisely the definition of symmetry for the associated relation.

71. Our picture is just like Figure 13, but with only three vertices on each side.



73. Suppose $P(i, j)$ and $P(k, l)$ need to communicate. Clearly by using $|i - k|$ hops we can move from $P(i, j)$ to $P(k, j)$. Then using $|j - l|$ hops we can move from $P(k, j)$ to $P(k, l)$. In all we used $|i - k| + |j - l|$ hops. But each of these absolute values is certainly less than m, since all the indices are less than m. Therefore the sum is less than $2m$, so it is $O(m)$.

SECTION 10.3 Representing Graphs and Graph Isomorphism

Human beings can get a good feeling for a small graph by looking at a picture of it drawn with points in the plane and lines or curves joining pairs of these points. If a graph is at all large (say with more than a dozen vertices or so), then the picture soon becomes too crowded to be useful. A computer has little use for nice pictures, no matter how small the vertex set. Thus people and machines need more precise—more discrete—representations of graphs. In this section we learned about some useful representations. They are for the most part exactly what any intelligent person would come up with, given the assignment to do so.

The only tricky idea in this section is the concept of graph isomorphism. It is a special case of a more general notion of isomorphism, or sameness, of mathematical objects in various settings. Isomorphism tries to capture the idea that all that really matters in a graph is the adjacency structure. If we can find a way to superimpose the graphs so that the adjacency structures match, then the graphs are, for all purposes that matter, the same. In trying to show that two graphs are isomorphic, try moving the vertices around in your mind to see whether you can make the graphs look the same. Of course there are often lots of things to help. For example, in every isomorphism, vertices that correspond must have the same degree.

A good general strategy for determining whether two graphs are isomorphic might go something like this. First check the degrees of the vertices to make sure there are the same number of each degree. See whether vertices of corresponding degrees follow the same adjacency pattern (e.g., if there is a vertex of degree 1 adjacent to a vertex of degree 4 in one of the graphs, then there must be the same pattern in the other, if the

graphs are isomorphic). Then look for triangles in the graphs, and see whether they correspond. Sometimes, if the graphs have lots of edges, it is easier to see whether the complements are isomorphic (see Exercise 46). If you cannot find a good reason for the graphs not to be isomorphic (an invariant on which they differ), then try to write down a one-to-one and onto function that shows them to be isomorphic (there may be more than one such function); such a function has to have vertices of like degrees correspond, so often the function practically writes itself. Then check each edge of the first graph to make sure that it corresponds to an edge of the second graph under this correspondence.

Unfortunately, no one has yet discovered a really good algorithm for determining graph isomorphism that works on all pairs of graphs. Research in this subject has been quite active in recent years. See Writing Project 10.

1. Adjacency lists are lists of lists. The adjacency list of an undirected graph is simply a list of the vertices of the given graph, together with a list of the vertices adjacent to each. The list for this graph is as follows. Since, for instance, b is adjacent to a and d, we list a and d in the row for b.

Vertex	Adjacent vertices
a	b, c, d
b	a, d
c	a, d
d	a, b, c

3. To form the adjacency list of a directed graph, we list, for each vertex in the graph, the terminal vertex of each edge that has the given vertex as its initial vertex. The list for this directed graph is as follows. For example, since there are edges from d to each of b, c, and d, we put those vertices in the row for d.

Initial vertex	Terminal vertices
a	a, b, c, d
b	d
c	a, b
d	b, c, d

5. For Exercises 5–8 we assume that the vertices are listed in alphabetical order. The matrix contains a 1 as entry (i, j) if there is an edge from vertex i to vertex j; otherwise that entry is 0.

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

7. This is similar to Exercise 5. Note that edges have direction here, so that, for example, the $(1, 2)$ entry is a 1 since there is an edge from a to b, but the $(2, 1)$ entry is a 0 since there is no edge from b to a. Also, the $(1, 1)$ entry is a 1 since there is a loop at a, but the $(2, 2)$ entry is a 0 since there is no loop at b.

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

9. We can solve these problems by first drawing the graph, then labeling the vertices, and finally constructing the matrix by putting a 1 in position (i, j) whenever vertices i and j are joined by an edge. It helps to choose a nice order, since then the matrix will have nice patterns in it.

a) The order of the vertices does not matter, since they all play the same role. The matrix has 0's on the diagonal, since there are no loops in the complete graph.

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

b) We put the vertex in the part by itself first.

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

c) We put the vertices in the part of size 2 first. Notice the block structure.

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

d) We put the vertices in the same order in the matrix as they are around the cycle.

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

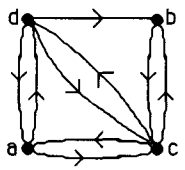
e) We put the center vertex first. Note that the last four columns of the last four rows represent a C_4.

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

f) We can label the vertices by the binary numbers from 0 to 7. Thus the first row (also the first column) of this matrix corresponds to the string 000, the second to the string 001, and so on. Since Q_3 has 8 vertices, this is an 8×8 matrix.

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

11. This graph has four vertices and is directed, since the matrix is not symmetric. We draw the four vertices as points in the plane, then draw a directed edge from vertex i to vertex j whenever there is a 1 in position (i, j) in the given matrix.



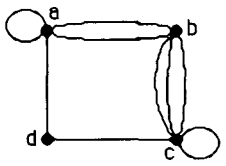
13. We use alphabetical order of the vertices for Exercises 13–15. If there are k parallel edges between vertices i and j, then we put the number k into the $(i,j)^{\text{th}}$ entry of the matrix. In this exercise, there is only one pair of parallel edges.

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 2 \\ 1 & 1 & 0 & 1 \\ 0 & 2 & 1 & 0 \end{bmatrix}$$

15. This is similar to Exercise 13. In this graph there are loops, which are represented by entries on the diagonal. For example, the loop at c is shown by the 1 as the $(3,3)^{\text{th}}$ entry.

$$\begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{bmatrix}$$

17. Because of the numbers larger than 1, we need multiple edges in this graph.



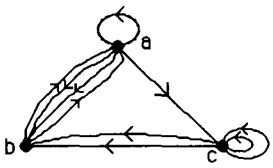
19. We use alphabetical order of the vertices. We put a 1 in position (i,j) if there is a directed edge from vertex i to vertex j; otherwise we make that entry a 0. Note that loops are represented by 1's on the diagonal.

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

21. This is similar to Exercise 19, except that there are parallel directed edges. If there are k parallel edges from vertex i to vertex j, then we put the number k into the $(i,j)^{\text{th}}$ entry of the matrix. For example, since there are 2 edges from a to c, the $(1,3)^{\text{th}}$ entry of the adjacency matrix is 2; the loop at c is shown by the 1 as the $(3,3)^{\text{th}}$ entry.

$$\begin{bmatrix} 1 & 1 & 2 & 1 \\ 1 & 0 & 0 & 2 \\ 1 & 0 & 1 & 1 \\ 0 & 2 & 1 & 0 \end{bmatrix}$$

23. Since the matrix is not symmetric, we need directed edges; furthermore, it must be a directed multigraph because of the entries larger than 1. For example, the 2 in position $(3,2)$ means that there are two parallel edges from vertex c to vertex b.



25. Since the matrix is symmetric, it has to be square, so it represents a graph of some sort. In fact, such a matrix does represent a simple graph. The fact that it is a zero-one matrix means that there are no parallel edges. The fact that there are 0's on the diagonal means that there are no loops. The fact that the matrix is symmetric means that the edges can be assumed to be undirected. Note that such a matrix also represents a directed graph in which all the edges happen to appear in antiparallel pairs (see the solution to Exercise 1d in Section 10.1 for a definition), but that is irrelevant to this question; the answer to the question asked is "yes."

27. In an incidence matrix we have one column for each edge. We use alphabetical order of the vertices. Loops are represented by columns with one 1; other edges are represented by columns with two 1's. The order in which the columns are listed is immaterial.

Exercise 13
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Exercise 14
$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Exercise 15
$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

29. In an undirected graph, each edge incident to a vertex j contributes 1 in the j^{th} column; thus the sum of the entries in that column is just the number of edges incident to j. Another way to state the answer is that the sum of the entries is the degree of j minus the number of loops at j, since each loop contributes 2 to the degree count.

In a directed graph, each edge whose terminal vertex is j contributes 1 in the j^{th} column; thus the sum of the entries in that column is just the number of edges that have j as their terminal vertex. Another way to state the answer is that the sum of the entries is the in-degree of j.

31. Since each column represents an edge, the sum of the entries in the column is either 2, if the edge has 2 incident vertices (i.e., is not a loop), or 1 if it has only 1 incident vertex (i.e., is a loop).

33. **a)** The incidence matrix for K_n has n rows and $C(n,2)$ columns. For each i and j with $1 \leq i < j \leq n$, there is a column with 1's in rows i and j and 0's elsewhere.

b) The matrix looks like this, with n rows and n columns.

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 1 \end{bmatrix}$$

c) The matrix looks like the matrix for C_n, except with an extra row of 0's (which we have put at the end), since the vertex "in the middle" is not involved in the edges "around the outside," and n more columns for the "spokes." We show some extra space between the rim edge columns and the spoke columns; this is for

human convenience only and does not have any bearing on the matrix itself.

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 1 & \quad & 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 & 0 & & 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 0 & 0 & & 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 & & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 1 & 1 & & 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & & 1 & 1 & 1 & \cdots & 1 \end{bmatrix}$$

d) This matrix has $m + n$ rows and mn columns, one column for each pair (i, j) with $1 \leq i \leq m$ and $1 \leq j \leq n$. We have put in some extra spacing for readability of the pattern.

$$\begin{bmatrix} 1 & 1 & \cdots & 1 & \quad & 0 & 0 & \cdots & 0 & \quad & \cdots & \quad & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & & 1 & 1 & \cdots & 1 & & \cdots & & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots & \vdots & \ddots & \vdots & & \ddots & & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & & 0 & 0 & \cdots & 0 & & \cdots & & 1 & 1 & \cdots & 1 \\ & & & & & & & & & & & & & & & \\ 1 & 0 & \cdots & 0 & & 1 & 0 & \cdots & 0 & & \cdots & & 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & & 0 & 1 & \cdots & 0 & & \cdots & & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots & \vdots & \ddots & \vdots & & \ddots & & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & & 0 & 0 & \cdots & 1 & & \cdots & & 0 & 0 & \cdots & 1 \end{bmatrix}$$

35. These graphs are isomorphic, since each is the 5-cycle. One isomorphism is $f(u_1) = v_1$, $f(u_2) = v_3$, $f(u_3) = v_5$, $f(u_4) = v_2$, and $f(u_5) = v_4$.

37. These graphs are isomorphic, since each is the 7-cycle (this is just like Exercise 35).

39. These two graphs are isomorphic. One can see this visually—just imagine "moving" vertices u_1 and u_4 into the inside of the rectangle, thereby obtaining the picture on the right. Formally, one isomorphism is $f(u_1) = v_5$, $f(u_2) = v_2$, $f(u_3) = v_3$, $f(u_4) = v_6$, $f(u_5) = v_4$, and $f(u_6) = v_1$.

41. These graphs are not isomorphic. In the first graph the vertices of degree 3 are adjacent to a common vertex. This is not true of the second graph.

43. These are isomorphic. One isomorphism is $f(u_1) = v_1$, $f(u_2) = v_9$, $f(u_3) = v_4$, $f(u_4) = v_3$, $f(u_5) = v_2$, $f(u_6) = v_8$, $f(u_7) = v_7$, $f(u_8) = v_5$, $f(u_9) = v_{10}$, and $f(u_{10}) = v_6$.

45. We must show that being isomorphic is reflexive, symmetric, and transitive. It is reflexive since the identity function from a graph to itself provides the isomorphism (the one-to-one correspondence)—certainly the identity function preserves adjacency and nonadjacency. It is symmetric, since if f is a one-to-one correspondence that makes G_1 isomorphic to G_2, then f^{-1} is a one-to-one correspondence that makes G_2 isomorphic to G_1; that is, f^{-1} is a one-to-one and onto function from V_2 to V_1 such that c and d are adjacent in G_2 if and only if $f^{-1}(c)$ and $f^{-1}(d)$ are adjacent in G_1. It is transitive, since if f is a one-to-one correspondence that makes G_1 isomorphic to G_2, and g is a one-to-one correspondence that makes G_2 isomorphic to G_3, then $g \circ f$ is a one-to-one correspondence that makes G_1 isomorphic to G_3.

47. If a vertex is isolated, then it has no adjacent vertices. Therefore in the adjacency matrix the row and column for that vertex must contain all 0's.

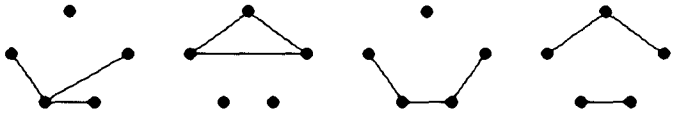
49. Let V_1 and V_2 be the two parts, say of sizes m and n, respectively. We can number the vertices so that all the vertices in V_1 come before all the vertices in V_2. The adjacency matrix has $m+n$ rows and $m+n$ columns. Since there are no edges between two vertices in V_1, the first m columns of the first m rows must all be 0's. Similarly, since there are no edges between two vertices in V_2, the last n columns of the last n rows must all be 0's. This is what we were asked to prove.

51. There are two such graphs, which can be found by trial and error. (We need only look for graphs with 5 vertices and 5 edges, since a self-complementary graph with 5 vertices must have $C(5,2)/2 = 5$ edges. If nothing else, we can draw them all and find the complement of each. See the pictures for the solution of Exercise 47d in Section 10.4.) One such graph is C_5. The other consists of a triangle, together with an edge from one vertex of the triangle to the fourth vertex, and an edge from another vertex of the triangle to the fifth vertex.

53. If C_n is to be self-complementary, then C_n must have the same number of edges as its complement. We know that C_n has n edges. Its complement has the number of edges in K_n minus the number of edges in C_n, namely $C(n,2)-n = [n(n-1)/2]-n$. If we set these two quantities equal we obtain $[n(n-1)/2]-n = n$, which has $n = 5$ as its only solution. Thus C_5 is the only C_n that *might* be self-complementary—our argument just shows that it has the same number of edges as its complement, not that it is indeed isomorphic to its complement. However, it we draw C_5 and then draw its complement, then we see that the complement is again a copy of C_5. Thus $n = 5$ is the answer to the problem.

55. We need to enumerate these graphs carefully to make sure of getting them all—leaving none out and not duplicating any. Let us organize our catalog by the degrees of the vertices. Since there are only 3 edges, the largest the degree could be is 3, and the only graph with 5 vertices, 3 edges, and a vertex of degree 3 is a $K_{1,3}$ together with an isolated vertex. If all the vertices that are not isolated have degree 2, then the graph must consist of a C_3 and 2 isolated vertices. The only way for there to be two vertices of degree 2 (and therefore also 2 of degree 1) is for the graph to be three edges strung end to end, together with an isolated vertex. The only other possibility is for 2 of the edges to be adjacent and the third to be not adjacent to either of the others. All in all, then, we have the 4 possibilities shown below.

See [ReWi] for more information about graph enumeration problems of this sort (such as Exercises 54, 56, and 68 in this section, Exercise 47 in Section 10.4, and supplementary exercises 2, 31, 32, and 40).



57. **a)** Both graphs consist of 2 sides of a triangle; they are clearly isomorphic.

b) The graphs are not isomorphic, since the first has 4 edges and the second has 5 edges.

c) The graphs are not isomorphic, since the first has 4 edges and the second has 3 edges.

59. There are at least two approaches we could take here. One approach is to have a correspondence not only of the vertices but also of the edges, with incidence (and nonincidence) preserved. In detail, we say that two pseudographs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there are one-to-one and onto functions $f : V_1 \to V_2$ and $g : E_1 \to E_2$ such that for each vertex $v \in V_1$ and edge $e \in E_1$, v is incident to e if and only if $f(v)$ is incident to $g(e)$.

Another approach is simply to count the number of edges between pairs of vertices. Thus we can define $G_1 = (V_1, E_1)$ to be isomorphic to $G_2 = (V_2, E_2)$ if there is a one-to-one and onto function $f : V_1 \to V_2$ such that for every pair of (not necessarily distinct) vertices u and v in V_1, there are exactly the same number of

edges in E_1 with $\{u, v\}$ as their set of endpoints as there are edges in E_2 with $\{f(u), f(v)\}$ as their set of endpoints.

61. We can tell by looking at the loop, the parallel edges, and the degrees of the vertices that if these directed graphs are to be isomorphic, then the isomorphism has to be $f(u_1) = v_3$, $f(u_2) = v_4$, $f(u_3) = v_2$, and $f(u_4) = v_1$. We then need to check that each directed edge (u_i, u_j) corresponds to a directed edge $(f(u_i), f(u_j))$. We check that indeed it does for each of the 7 edges (and there are only 7 edges in the second graph). Therefore the two graphs are isomorphic.

63. If there is to be an isomorphism, the vertices with the same in-degree would have to correspond, and the edge between them would have to point in the same direction, so we would need u_1 to correspond to v_3, and u_2 to correspond to v_1. Similarly we would need u_3 to correspond to v_4, and u_4 to correspond to v_2. If we check all 6 edges under this correspondence, then we see that adjacencies are preserved (in the same direction), so the graphs are isomorphic.

65. If f is an isomorphism from a directed graph G to a directed graph H, then f is also an isomorphism from G^c to H^c. This is clear, because (u, v) is an edge of G^c if and only if (v, u) is an edge of G if and only if $(f(v), f(u))$ is an edge of H if and only if $(f(u), f(v))$ is an edge of H^c.

67. A graph with a triangle will not be bipartite, but cycles of even length are bipartite. So we could let one graph be C_6 and the other be the union of two disjoint copies of C_3.

69. Suppose that the graph has v vertices and e edges. Then the incidence matrix is a $v \times e$ matrix, so its transpose is an $e \times v$ matrix. Therefore the product is a $v \times v$ matrix. Suppose that we denote the typical entry of this product by a_{ij}. Let t_{ik} be the typical entry of the incidence matrix; it is either a 0 or a 1. By definition

$$a_{ij} = \sum_{k=1}^{e} t_{ik} t_{jk} .$$

We can now read off the answer from this equation. If $i \neq j$, then a_{ij} is just a count of the number of edges incident to both i and j—in other words, the number of edges between i and j. On the other hand a_{ii} is equal to the number of edges incident to i.

71. Perhaps the simplest example would be to have the graphs have all degrees equaling 2. One way for this to happen is for the graph to be a cycle. But it will also happen if the graph is a disjoint union of cycles. The smallest example occurs when there are six vertices. If G_1 is the 6-cycle and G_2 is the union of two triangles, then the degree sequences are $(2, 2, 2, 2, 2, 2)$ for both, but obviously the graphs are not isomorphic. If we want a connected example, then look at Exercise 41, where the degree sequence is $(3, 3, 2, 2, 1, 1, 1, 1)$ for each graph.

SECTION 10.4 Connectivity

Some of the most important uses of graphs deal with the notion of path, as the examples and exercises in this and subsequent sections show. It is important to understand the definitions, of course. Many of the exercises here are straightforward. The reader who wants to get a better feeling for what the arguments in more advanced graph theory are like should tackle problems like Exercises 35–38.

1. **a)** This is a path of length 4, but it is not simple, since edge $\{b, c\}$ is used twice. It is not a circuit, since it ends at a different vertex from the one at which it began.

b) This is not a path, since there is no edge from c to a.

c) This is not a path, since there is no edge from b to a.

d) This is a path of length 5 (it has 5 edges in it). It is simple, since no edge is repeated. It is a circuit since it ends at the same vertex at which it began.

3. This graph is not connected—it has three components.

5. This graph is not connected. There is no path from the vertices in one of the triangles to the vertices in the other.

7. A connected component of an acquaintanceship graph represent a maximal set of people with the property that for any two of them, we can find a string of acquaintances that takes us from one to the other. The word "maximal" here implies that nobody else can be added to this set of people without destroying this property.

9. If a person has Erdős number n, then there is a path of length n from that person to Erdős in the collaboration graph. By definition, that means that that person is in the same component as Erdős. Conversely, if a person is in the same component as Erdős, then there is a path from that person to Erdős, and the length of a shortest such path is that person's Erdős number.

11. **a)** Notice that there is no path from a to any other vertex, because both edges involving a are directed toward a. Therefore the graph is not strongly connected. However, the underlying undirected graph is clearly connected, so this graph is weakly connected.

b) Notice that there is no path from c to any other vertex, because both edges involving c are directed toward c. Therefore the graph is not strongly connected. However, the underlying undirected graph is clearly connected, so this graph is weakly connected.

c) The underlying undirected graph is clearly not connected (one component has vertices b, f, and e), so this graph is neither strongly nor weakly connected.

13. The strongly connected components are the maximal sets of phone numbers for which it is possible to find directed paths between every two different numbers in the set, where the existence of a directed path from phone number x to another phone number y means that x called some number, which called another number, ..., which called y. (The number of intermediary phone numbers in this path can be any natural number.)

15. In each case we want to look for large sets of vertices all which of which have paths to all the others. For these graphs, this can be done by inspection. These will be the strongly connected components.

a) Clearly $\{a, b, f\}$ is a set of vertices with paths between all the vertices in the set. The same can be said of $\{c, d, e\}$. Every edge between a vertex in the first set and a vertex in the second set is directed from the first, to the second. Hence there are no paths from c, d, or e to a, b, or f, and therefore these vertices are not in the same strongly connected component. Therefore these two sets are the strongly connected component.

b) The circuits a, e, d, c, b, a and a, e, d, h, a show that these six vertices are all in the same component. There is no path from f to any of these vertices, and no path from g to any other vertex. Therefore f and g are not in the same strong component as any other vertex. Therefore the strongly connected components are $\{a, b, c, d, e, h\}$, $\{f\}$, and $\{g\}$.

c) It is clear that a and i are in the same strongly connected component. If we look hard, we can also find the circuit b, h, f, g, d, e, d, b, so these vertices are in the same strongly connected component. Because of edges ig and hi, we can get from either of these collections to the other. Thus $\{a, b, d, e, f, g, h, i\}$ is a strong component. We cannot travel from c to any other vertex, so c is in a component by itself.

17. The hardest part of this exercise is figuring out what we need to prove. It is enough to prove that if the strong components of u and v are not disjoint then they are the same. So suppose that w is a vertex that is in both the strong component of u and the strong component of v. (It is enough to consider the vertices in these components, because the edges in a strong component are just all the edges joining the vertices in that component.) This means that there are directed paths (in each direction) between u and w and between v and w. It follows that there are directed paths from u to v and from v to u, via w. Suppose x is a vertex in the strong component of u. Then x is also in the strong component of v, because there is a path from x to v (namely the path from x to u followed by the path from u to v) and vice versa.

19. One approach here is simply to invoke Theorem 2 and take successive powers of the adjacency matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

The answers are the off-diagonal elements of these powers. An alternative approach is to argue combinatorially as follows. Without loss of generality, we assume that the vertices are called $1, 2, 3, 4$, and the path is to run from 1 to 2. A path of length n is determined by choosing the $n-1$ intermediate vertices. Each vertex in the path must differ from the one immediately preceding it.

a) A path of length 2 requires the choice of 1 intermediate vertex, which must be different from both of the ends. Vertices 3 and 4 are the only ones available. Therefore the answer is 2.

b) Let the path be denoted $1, x, y, 2$. If $x = 2$, then there are 3 choices for y. If $x = 3$, then there are 2 choices for y; similarly if $x = 4$. Therefore there are $3 + 2 + 2 = 7$ possibilities in all.

c) Let the path be denoted $1, x, y, z, 2$. If $x = 3$, then by part **(b)** there are 7 choices for y and z. Similarly if $x = 4$. If $x = 2$, then y and z can be any two distinct members of $\{1, 3, 4\}$, and there are $P(3, 2) = 6$ ways to choose them. Therefore there are $7 + 7 + 6 = 20$ possibilities in all.

d) Let the path be denoted $1, w, x, y, z, 2$. If $w = 3$, then by part **(c)** there are 20 choices for x, y, and z. Similarly if $w = 4$. If $w = 2$, then x must be different from 2, and there are 3 choices for x. For each of these there are by part **(b)** 7 choices for y and z. This gives a total of 21 possibilities in this case. Therefore the answer is $20 + 20 + 21 = 61$.

21. Graph G has a triangle (u_1, u_2, u_3). Graph H does not (in fact, it is bipartite). Therefore G and H are not isomorphic.

23. The drawing of G clearly shows it to be the cube Q_3. Can we see H as a cube as well? Yes—we can view the outer ring as the top face, and the inner ring as the bottom face. We can imagine walking around the top face of G clockwise (as viewed from above), then dropping down to the bottom face and walking around it counterclockwise, finally returning to the starting point on the top face. This is the path $u_1, u_2, u_7, u_6, u_5, u_4, u_3, u_8, u_1$. The corresponding path in H is $v_1, v_2, v_3, v_4, v_5, v_8, v_7, v_6, v_1$. We can verify that the edges not in the path do connect corresponding vertices. Therefore $G \cong H$.

25. As explained in the solution to Exercise 19, we could take powers of the adjacency matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

The answers are found in location $(1, 2)$, for instance. Using the alternative approach is much easier than in Exercise 19. First of all, two nonadjacent vertices must lie in the same part, so only paths of even length can

join them. Also, there are clearly 3 choices for each intermediate vertex in a path. Therefore we have the following answers:

a) $3^1 = 3$ **b)** 0 **c)** $3^3 = 27$ **d)** 0

27. There are two approaches here. We could use matrix multiplication on the adjacency matrix of this directed graph (by Theorem 2), which is

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

Thus we can compute $\mathbf{A}^2$ for part **(a)**, $\mathbf{A}^3$ for part **(b)**, and so on, and look at the $(1,5)^{\text{th}}$ entry to determine the number of paths from a to e. Alternately, we can argue in an ad hoc manner, as we do below.

a) There is just 1 path of length 2, namely a, b, e.

b) There are no paths of length 3, since after 3 steps, a path starting at a must be at b, c, or d.

c) For a path of length 4 to end at e, it must be at b after 3 steps. There are only 2 such paths, a, b, a, b, e and a, d, a, b, e.

d) The only way for a path of length 5 to end at e is for the path to go around the triangle bec. Therefore only the path a, b, e, c, b, e is possible.

e) There are several possibilities for a path of length 6. Since the only way to get to e is from b, we are asking for the number of paths of length 5 from a to b. We can go around the square (a, b, e, d, a, b), or else we can jog over to either b or d and back twice—there being 4 ways to choose where to do the jogging. Therefore there are 5 paths in all.

f) As in part **(d)**, it is clear that we have to use the triangle. We can either have a, b, a, b, e, c, b, e or a, d, a, b, e, c, b, e or a, b, e, c, b, a, b, e. Thus there are 3 paths.

29. The definition given here makes it clear that u and v are related if and only if they are in the same component—in other words $f(u) = f(v)$ where $f(x)$ is the component in which x lies. Therefore by Exercise 9 in Section 9.5 this is an equivalence relation.

31. A cut vertex is one whose removal splits the graph into more components than it originally had (which is 1 in this case). Only vertex c is a cut vertex here. If it is removed, then the resulting graph will have two components. If any other vertex is removed, then the graph remains connected.

33. There are several cut vertices here: b, c, e, and i. Removing any of these vertices creates a graph with more than one component. The removal of any of the other vertices leaves a graph with just one component.

35. Without loss of generality, we can restrict our attention to the component in which the cut edge lies; other components of the graph are irrelevant to this proposition. To fix notation, let the cut edge be uv. When the cut edge is removed, the graph has two components, one of which contains v and the other of which contains u. If v is pendant, then it is clear that the removal of v results in exactly the component containing u—a connected graph. Therefore v is not a cut vertex in this case. On the other hand, if v is not pendant, then there are other vertices in the component containing v—at least one other vertex w adjacent to v. (We are assuming that this proposition refers to a simple graph, so that there is no loop at v.) Therefore when v is removed, there are at least two components, one containing u and another containing w.

37. If every component of G is a single vertex, then clearly no vertex is a cut vertex (the removal of any of them actually decreases the number of components rather than increasing it). Therefore we may as well assume

that some component of G has at least two vertices, and we can restrict our attention to that component; in other words, we can assume that G is connected. One clever way to do this problem is as follows. Define the **distance** between two vertices u and v, denoted $d(u,v)$, to be the length of a shortest path joining u and v. Now choose u and v so that $d(u,v)$ is as large as possible. We claim that neither u nor v is a cut vertex. Suppose otherwise, say that u is a cut vertex. Then v is in one component that results after u is removed, and some vertex w is in another. Since there is no path from w to v in the graph with u removed, every path from w to v must have passed through u. Therefore the distance between w and v must have been strictly greater than the distance between u and v. This is a contradiction to the choice of u and v, and our proof by contradiction is complete.

39. This problem is simply asking for the cut edges of these graphs.

a) The link joining Denver and Chicago and the link joining Boston and New York are the cut edges.

b) The following links are the cut edges: Seattle–Portland, Portland–San Francisco, Salt Lake City–Denver, New York–Boston, Boston–Bangor, Boston–Burlington.

41. A vertex basis will be a set of people who collectively can influence everyone, at least indirectly, but none of whom influences another member of that set (otherwise the set would not be minimal). The set consisting of Deborah is a vertex basis, since she can influence everyone except Yvonne directly, and she can influence Yvonne indirectly through Brian.

43. Since there can be no edges between vertices in different components, G will have the most edges when each of the components is a complete graph. Since K_{n_i} has $C(n_i, 2)$ edges, the maximum number of edges is the sum given in the exercise.

45. Before we give a correct proof here, let us look at an incorrect proof that students often give for this exercise. It goes something like this. "Suppose that the graph is not connected. Then no vertex can be adjacent to every other vertex, only to $n-2$ other vertices. One vertex joined to $n-2$ other vertices creates a component with $n-1$ vertices in it. To get the most edges possible, we must use all the edges in this component. The number of edges in this component is thus $C(n-1,2) = (n-1)(n-2)/2$, and the other component (with only one vertex) has no edges. Thus we have shown that a disconnected graph has at most $(n-1)(n-2)/2$ edges, so every graph with more edges than that has to be connected." The fallacy here is in assuming—without justification—that the maximum number of edges is achieved when one component has $n-1$ vertices. What if, say, there were two components of roughly equal size? Might they not together contain more edges? We will see that the answer is "no," but it is important to realize that this requires proof—it is not obvious without some calculations.

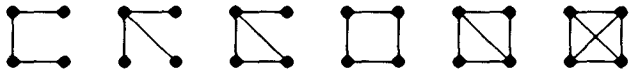
Here is a correct proof, then. Suppose that the graph is not connected. Then it has a component with k vertices in it, for some k between 1 and $n-1$, inclusive. The remaining $n-k$ vertices are in one or more other components. The maximum number of edges this graph could have is then $C(k,2) + C(n-k,2)$, which, after a bit of algebra, simplifies to $k^2 - nk + (n^2 - n)/2$. This is a quadratic function of k. It is minimized when $k = n/2$ (the k coordinate of the vertex of the parabola that is the graph of this function) and maximized at the endpoints of the domain, namely $k = 1$ and $k = n-1$. In the latter cases its value is $(n-1)(n-2)/2$. Therefore the largest number of edges that a disconnected graph can have is $(n-1)(n-2)/2$, so every graph with more edges than this must be connected.

47. We have to enumerate carefully all the possibilities.

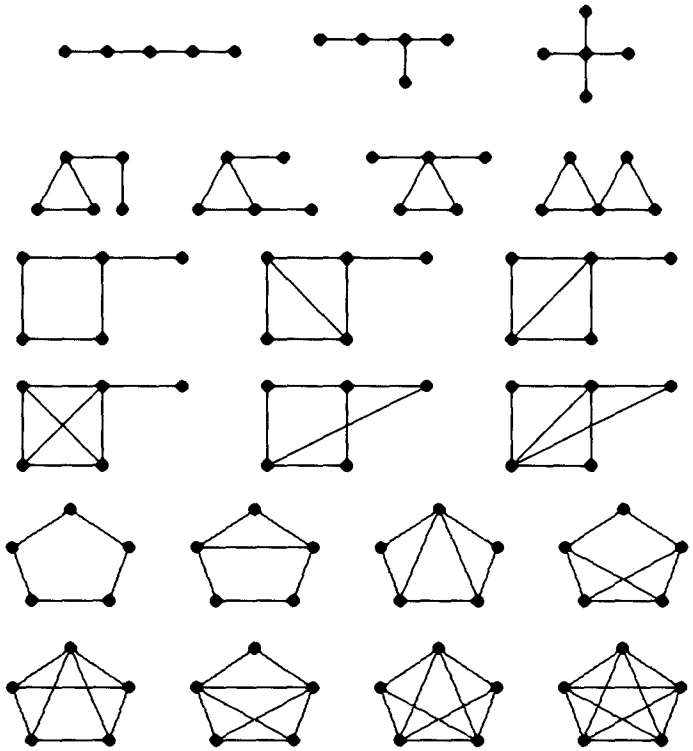
a) There is obviously only 1, namely K_2, the graph consisting of two vertices and the edge between them.

b) There are clearly 2 connected graphs with 3 vertices, namely K_3 and K_3 with one edge deleted, as shown.

c) There are several connected graphs with $n = 4$. If the graph has no circuits, then it must either be a path of length 3 or the "star" $K_{1,3}$. If it contains a triangle but no copy of C_4, then the other vertex must be pendant—only 1 possibility. If it contains a copy of C_4, then neither, one, or both of the other two edges may be present—3 possibilities. Therefore the answer is $2 + 1 + 3 = 6$. The graphs are shown below.



d) We need to enumerate the possibilities in some systematic way, such as by the largest cycle contained in the graph. There are 21 such graphs, as can be seen by such an enumeration, shown below. First we show those graphs with no circuits, then those with a triangle but no C_4 or C_5, then those with a C_4 but no C_5, and finally those with a C_5. In doing this problem we have to be careful not only not to leave out any graphs, but also not to list any twice.



49. In each case we just need to verify that the removal of an edge will not disconnect the graph.

a) Removing an edge from a cycle leaves a path, which is still connected.

b) Removing an edge from the cycle portion of the wheel leaves that portion still connected as in part **(a)**, and the central vertex is clearly still connected to it as well. Removing a spoke leaves the cycle intact and the central vertex still connected to it as well.

c) Let u, v, a, b be any four vertices of $K_{m,n}$ with u and v in one part and a and b in the other. They are connected by the 4-cycle $uavb$. Removing one edge will not disconnect this 4-cycle, so these vertices are still connected, and the entire graph is therefore still connected. Note that we needed $m, n \geq 2$ for this to work (and for the statement to be true).

d) Think of Q_n as two copies of Q_{n-1} with corresponding vertices joined by an edge. Without loss of generality we can assume that the removed edge is one of the edges joining corresponding vertices. Since each Q_{n-1} is connected and at least one edge remains joining the two copies, the resulting graph is connected.

51. If G is complete, then removing vertices one by one leaves a complete graph at each step, so we never get a disconnected graph. Conversely, if G is not complete, say with edge uv missing, then removing all the vertices except u and v creates the disconnected graph consisting of just those two vertices.

53. Without loss of generality, assume $m \leq n$. We can disconnect $K_{m,n}$ by removing the m vertices in the smaller part. To see that removing fewer than m vertices will not disconnect the graph, note that given any two vertices u and v, there are m paths that pairwise share nothing except their endpoints; these paths are of length 2 if u and v are in the same part and of length 1 or 3 if they are in different parts. Removing fewer than m vertices can cut at most $m-1$ of these paths, so the resulting graph is still connected. Therefore $\kappa(K_{m,n}) = \min(m,n)$. It is also clear that $\lambda(K_{m,n}) \leq \min(m,n)$, because we can disconnect the graph by removing all the edges incident to a vertex in the larger part. By the inequality stated after Example 9, $\lambda(K_{m,n}) \geq \kappa(K_{m,n})$. Therefore $\lambda(K_{m,n}) = \min(m,n)$ as well.

55. Let G be a graph with n vertices. Note that $\kappa(G) \leq n-1$. Suppose a smallest edge cut C (i.e., one with $|C| = \lambda(G)$) leaves a nonempty proper subset S of the vertices of G disconnected from the complementary set $S' = V - S$. If xy is an edge of G for every $x \in S$ and $y \in S'$, then the size of C is $|S||S'|$, which is at least $n-1$ (it is this small only if $|S| = 1$ or $n-1$), so $\kappa(G) \leq \lambda(G)$ in this case. Otherwise, let $x \in S$ and $y \in S'$ be nonadjacent vertices. Let T consist of all neighbors of x in S' together with all vertices of $S - \{x\}$ with neighbors in S'. Then T is a vertex cut, because it separates x and y. Now look at the edges from x to $T \cap S'$ and one edge from each vertex of $T \cap S$ to S'; this gives us $|T|$ distinct edges that lie in C, so $\lambda(G) = |C| \geq |T| \geq \kappa(G)$.

57. We need to look at successive powers of the adjacency matrix until we find one in which the $(1,6)^{\text{th}}$ entry is not 0. Since the matrix is

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix},$$

we see that the $(1,6)^{\text{th}}$ entry of $\mathbf{A}^2$ is 2. Thus there is a path of length 2 from a to f (in fact 2 of them). On the other hand there is no path of length 1 from a to f (i.e., no edge), so the length of a shortest path is 2.

59. Let the simple paths P_1 and P_2 be $u = x_0, x_1, \ldots, x_n = v$ and $u = y_0, y_1, \ldots, y_m = v$, respectively. The paths thus start out at the same vertex. Since the paths do not contain the same set of edges, they must diverge eventually. If they diverge only after one of them has ended, then the rest of the other path is a simple circuit from v to v. Otherwise we can suppose that $x_0 = y_0$, $x_1 = y_1$, $\ldots$, $x_i = y_i$, but $x_{i+1} \neq y_{i+1}$. To form our simple circuit, we follow the path y_i, y_{i+1}, y_{i+2}, and so on, until it once again first encounters a vertex on P_1 (possibly as early as y_{i+1}, no later than y_m). Once we are back on P_1, we follow it along—forwards or backwards, as necessary—to return to x_i. Since $x_i = y_i$, this certainly forms a circuit. It must be a simple circuit, since no edge among the x_k's or the y_l's can be repeated (P_1 and P_2 are simple by hypothesis) and no edge among the x_k's can equal one of the edges y_l that we used, since we abandoned P_2 for P_1 as soon as we hit P_1.

61. Let $\mathbf{A}$ be the adjacency matrix of a given graph G. Theorem 2 tells us that $\mathbf{A}^r$ counts the number of paths of length r between vertices. If an entry in $\mathbf{A}^r$ is greater than 0, then there is a path between the corresponding vertices in G. Suppose that we look at $\mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \cdots + \mathbf{A}^{n-1}$, where n is the number of vertices in G. If there is a path between a pair of distinct vertices in G, then there is a path of length at most $n-1$, so this sum

will have a positive integer in the corresponding entry. Conversely, if there is no path, then the corresponding entry in every summand will be 0, and hence the entry in the sum will be 0. Therefore the graph is connected (i.e., there is a path between every pair of distinct vertices in G) if and only if every off-diagonal entry in this sum is strictly positive. To determine whether G is connected, therefore, we just compute this sum and check to see whether this condition holds.

63. We have to prove a statement and its converse here. One direction is fairly easy. If the graph is bipartite, say with parts A and B, then the vertices in every path must alternately lie in A and B. Therefore a path that starts in A, say, will end in B after an odd number of steps and in A after an even number of steps. Since a circuit ends at the same vertex where it starts, the length must be even. The converse is a little harder. We suppose that all circuits have even length and want to show that the graph is bipartite. We can assume that the graph is connected, because if it is not, then we can just work on one component at a time. Let v be a vertex of the graph, and let A be the set of all vertices to which there is a path of odd length starting at v, and let B be the set of all vertices to which there is a path of even length starting at v. Since the component is connected, every vertex lies in A or B. No vertex can lie in both A and B, since then following the odd-length path from v to that vertex and then back along the even-length path from that vertex to v would produce an odd circuit, contrary to the hypothesis. Thus the set of vertices has been partitioned into two sets. Now we just need to show that every edge has endpoints in different parts. If xy is an edge where $x \in A$, then the odd-length path from v to x followed by xy produces an even-length path from v to y, so $y \in B$ (and similarly if $x \in B$).

65. Suppose the couples are Bob and Carol Sanders, and Ted and Alice Henderson (these were characters in a movie from 1969). We represent the initial position by $(BCTA\bullet, \emptyset)$, indicating that all four people are on the left shore along with the boat (the dot). We want to reach the position $(\emptyset, BCTA\bullet)$. Positions will be the vertices of our graph, and legal transitions will be the edges. If Bob and Carol take the boat over, then we reach the position $(TA, BC\bullet)$. The only useful transition at that point is for someone to row back. Let's try Bob; so we have $(BTA\bullet, C)$. If Bob and Ted now row to the right shore, we reach $(A, BCT\bullet)$. Ted can take the boat back to fetch his wife, giving us $(TA\bullet, BC)$ and then $(\emptyset, BCTA\bullet)$. Notice that this path never violates the jealousy conditions imposed in this problem. The entire graph model would have many more positions, but we just need one path.

SECTION 10.5 Euler and Hamilton Paths

An Euler circuit or Euler path uses every edge exactly once. A Hamilton circuit or Hamilton path uses every vertex exactly once (not counting the circuit's return to its starting vertex). Euler and Hamilton circuits and paths have an important place in the history of graph theory, and as we see in this section they have some interesting applications. They provide a nice contrast—there are good algorithms for finding Euler paths (see also Exercises 50–53), but computer scientists believe that there is no good (efficient) algorithm for finding Hamilton paths.

Most of these exercises are straightforward. The reader should at least look at Exercises 16 and 17 to see how the concept of Euler path applies to directed graphs—these exercises are not hard if you understood the proof of Theorem 1 (given in the text before the statement of the theorem).

1. Since there are four vertices of odd degree (a, b, c, and e) and $4 > 2$, this graph has neither an Euler circuit nor an Euler path.

3. Since there are two vertices of odd degree (a and d), this graph has no Euler circuit, but it does have an Euler path starting at a and ending at d. We can find such a path by inspection, or by using the splicing idea explained in this section. One such path is $a, e, c, e, b, e, d, b, a, c, d$.

5. All the vertex degrees are even, so there is an Euler circuit. We can find such a circuit by inspection, or by using the splicing idea explained in this section. One such circuit is $a, b, c, d, c, e, d, b, e, a, e, a$.

7. All the vertex degrees are even, so there is an Euler circuit. We can find such a circuit by inspection, or by using the splicing idea explained in this section. One such circuit is $a, b, c, d, e, f, g, h, i, a, h, b, i, c, e, h, d, g, c, a$.

9. No, an Euler circuit does not exist in the graph modeling this hypothetical city either. Vertices A and B have odd degree.

11. Assuming we have just one truck to do the painting, the truck must follow an Euler path through the streets in order to do the job without traveling a street twice. Therefore this can be done precisely when there is an Euler path or circuit in the graph, which means that either zero or two vertices (intersections) have odd degree (number of streets meeting there). We are assuming, of course, that the city is connected.

13. In order for the picture to be drawn under the conditions of Exercises 13–15, the graph formed by the picture must have an Euler path or Euler circuit. Note that all of these graphs are connected. The graph in the current exercise has all vertices of even degree; therefore it has an Euler circuit and can be so traced.

15. See the comments in the solution to Exercise 13. This graph has 4 vertices of odd degree; therefore it has no Euler path or circuit and cannot be so traced.

17. If there is an Euler path, then as we follow it through the graph, each vertex except the starting and ending vertex must have equal in-degree and out-degree, since whenever we come to the vertex along some edge, we leave it along some edge. The starting vertex must have out-degree 1 greater than its in-degree, since after we have started, using one edge leading out of this vertex, the same argument applies. Similarly, the ending vertex must have in-degree 1 greater than its out-degree, since until we end, using one edge leading into this vertex, the same argument applies. Note that the Euler path itself guarantees weak connectivity; given any two vertices, there is a path from the one that occurs first along the Euler path to the other, via the Euler path.

Conversely, suppose that the graph meets the degree conditions stated here. By Exercise 16 it cannot have an Euler circuit. If we add one more edge from the vertex of deficient out-degree to the vertex of deficient in-degree, then the graph now has every vertex with its in-degree equal to its out-degree. Certainly the graph is still weakly connected. By Exercise 16 there is an Euler circuit in this new graph. If we delete the added edge, then what is left of the circuit is an Euler path from the vertex of deficient in-degree to the vertex of deficient out-degree.

19. For Exercises 18–23 we use the results of Exercises 16 and 17. By Exercise 16, we cannot hope to find an Euler circuit since vertex b has different out-degree and in-degree. By Exercise 17, we cannot hope to find an Euler path since vertex b has out-degree and in-degree differing by 2.

21. This directed graph satisfies the condition of Exercise 17 but not that of Exercise 16. Therefore there is no Euler circuit. The Euler path must go from a to e. One such path is $a, d, e, d, b, a, e, c, e, b, c, b. e$.

23. There are more than two vertices whose in-degree and out-degree differ by 1, so by Exercises 16 and 17, there is no Euler path or Euler circuit.

25. The algorithm is very similar to Algorithm 1. The input is a weakly connected directed multigraph in which either each vertex has in-degree equal to its out-degree, or else all vertices except two satisfy this condition and the remaining vertices have in-degree differing from out-degree by 1 (necessarily once in each direction). We begin by forming a path starting at the vertex whose out-degree exceeds its in-degree by 1 (in the second case) or at any vertex (in the first case). We traverse the edges (never more than once each), forming a path, until we cannot go on. Necessarily we end up either at the vertex whose in-degree exceeds its out-degree (in the first case) or at the starting vertex (in the second case). From then on we do exactly as in Algorithm 1, finding a simple circuit among the edges not yet used, starting at any vertex on the path we already have; such a vertex exists by the weak connectivity assumption. We splice this circuit into the path, and repeat the process until all edges have been used.

27. **a)** Clearly K_2 has an Euler path but no Euler circuit. For odd $n > 2$ there is an Euler circuit (since the degrees of all the vertices are $n-1$, which is even), whereas for even $n > 2$ there are at least 4 vertices of odd degree and hence no Euler path. Thus for no n other than 2 is there an Euler path but not an Euler circuit.
b) Since C_n has an Euler circuit for all n, there are no values of n meeting these conditions.
c) A wheel has at least 3 vertices of degree 3 (around the rim), so there can be no Euler path.
d) The same argument applies here as applied in part **(a)**. In more detail, Q_1 (which is the same as K_2) is the only cube with an Euler path but no Euler circuit, since for odd $n > 1$ there are too many vertices of odd degree, and for even $n > 1$ there is an Euler circuit.

29. Just as a graph with 2 vertices of odd degree can be drawn with one continuous motion, a graph with $2m$ vertices of odd degree can be drawn with m continuous motions. The graph in Exercise 1 has 4 vertices of odd degree, so it takes 2 continuous motions; in other words, the pencil must be lifted once. We could do this, for example, by first tracing a, c, d, e, a, b and then tracing c, b, e. The graphs in Exercises 2–7 all have Euler paths, so no lifting is necessary.

31. It is clear that a, b, c, d, e, a is a Hamilton circuit.

33. There is no Hamilton circuit because of the cut edges ($\{c, e\}$, for instance). Once a purported circuit had reached vertex e, there would be nowhere for it to go.

35. There is no Hamiltonian circuit in this graph. If there were one, then it would have to include all the edges of the graph, because it would have to enter and exit vertex a, enter and exit vertex d, and enter and exit vertex e. But then vertex c would have been visited more than once, a contradiction.

37. This graph has the Hamilton path a, b, c, f, d, e. This simple path hits each vertex once.

39. This graph has the Hamilton path f, e, d, a, b, c.

41. There are eight vertices of degree 2 in this graph. Only two of them can be the end vertices of a Hamilton path, so for each of the other six their two incident edges must be present in the path. Now if either all four of the "outside" vertices of degree 2 (a, c, g, and e) or all four of the "inside" vertices of degree 2 (i, k,

l, and n) are not end vertices, then a circuit will be completed that does not include all the vertices—either the outside square or the middle square. Therefore if there is to be a Hamilton path then exactly one of the inside corner vertices must be an end vertex, and each of the other inside corner vertices must have its two incident edges in the path. Without loss of generality we can assume that vertex i is an end, and that the path begins i, o, n, m, l, q, k, j. At this point, either the path must visit vertex p, in which case it gets stuck, or else it must visit b, in which case it will never be able to reach p. Either case gives a contradiction, so there is no Hamilton path.

43. It is easy to write down a Hamiltonian path here; for example, a, d, g, h, i, f, c, e, b.

45. A Hamilton circuit in a bipartite graph must visit the vertices in the parts alternately, returning to the part in which it began. Therefore a necessary condition is certainly $m = n$. Furthermore $K_{1,1}$ does not have a Hamilton circuit, so we need $n \geq 2$ as well. On the other hand, since the complete bipartite graph has all the edges we need, these conditions are sufficient. Explicitly, if the vertices are $a_1, a_2, \ldots, a_n$ in one part and $b_1, b_2, \ldots, b_n$ in the other, with $n \geq 2$, then one Hamilton circuit is $a_1, b_1, a_2, b_2, \ldots, a_n, b_n, a_1$.

47. For Dirac's theorem to be applicable, we need every vertex to have degree at least $n/2$, where n is the number of vertices in the graph. For Ore's theorem, we need $\deg(x) + \deg(y) \geq n$ whenever x and y are not adjacent.

a) In this graph $n = 5$. Dirac's theorem does not apply, since there is a vertex of degree 2, and 2 is smaller than $n/2$. Ore's theorem also does not apply, since there are two nonadjacent vertices of degree 2, so the sum of their degrees is less than n. However, the graph does have a Hamilton circuit—just go around the pentagon. This illustrates that neither of the sufficient conditions for the existence of a Hamilton circuit given in these theorems is necessary.

b) Everything said in the solution to part **(a)** is valid here as well.

c) In this graph $n = 5$, and all the vertex degrees are either 3 or 4, both of which are at least $n/2$. Therefore Dirac's theorem guarantees the existence of a Hamilton circuit. Ore's theorem must apply as well, since $(n/2) + (n/2) = n$; in this case, the sum of the degrees of any pair of nonadjacent vertices (there are only two such pairs) is 6, which is greater than or equal to 5.

d) In this graph $n = 6$, and all the vertex degrees are 3, which is (at least) $n/2$. Therefore Dirac's theorem guarantees the existence of a Hamilton circuit. Ore's theorem must apply as well, since $(n/2) + (n/2) = n$; in this case, the sum of the degrees of any pair of nonadjacent vertices is 6.

Although not illustrated in any of the examples in this exercise, there are graphs for which Ore's theorem applies, even though Dirac's does not. Here is one: Take K_4, and then tack on a path of length 2 between two of the vertices, say a, b, c. In all, this graph has five vertices, two with degree 3, two with degree 4, and one with degree 2. Since there is a vertex with degree less than $5/2$, Dirac's theorem does not apply. However, the sum of the degrees of any two (nonadjacent) vertices is at least $2 + 3 = 5$, so Ore's theorem does apply and guarantees that there is a Hamilton circuit.

49. The trick is to use a Gray code for n to build one for $n + 1$. We take the Gray code for n and put a 0 in front of each term to get half of the Gray code for $n + 1$; we put a 1 in front to get the second half. Then we reverse the second half so that the junction at which the two halves meet differ in only the first bit. For a formal proof we use induction on n. For $n = 1$ the code is $0, 1$ (which is not really a Hamilton circuit in Q_1). Assume the inductive hypothesis that $c_1, c_2, \ldots, c_{2^n}$ is a Gray code for n. Then $0c_1, 0c_2, \ldots, 0c_{2^n}, 1c_{2^n}, \ldots, 1c_2, 1c_1$ is a Gray code for $n + 1$.

51. Turning this verbal description into pseudocode is straightforward, especially if we allow ourselves lots of words in the pseudocode. We build our *circuit* (which we think of simply as an ordered list of edges) one edge at a time, keeping track of the vertex v we are at; the subgraph containing the edges we have not yet used we will

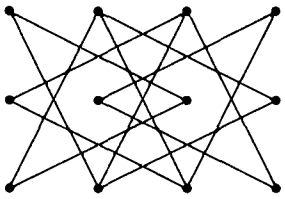
call H. We assume that the vertices of G are listed in some order, so that when we are asked to choose an edge from v meeting certain conditions, we can choose the edge to the vertex that comes first in this order among all those edges meeting the conditions. (This avoids ambiguity, which an algorithm is not supposed to have.)

procedure *fleury*(G : connected multigraph with all degrees even)
$v :=$ first vertex of G
circuit := the empty circuit
$H := G$
while H has edges
    Let e be an edge in H with v as one of its endpoints,
        such that e is not a cut edge of H, if such an edge
        exists; otherwise let e be any edge in H with v as
        one of its endpoints.
    $v :=$ other endpoint of e
    Add e to the end of *circuit*
    Remove e from H
return *circuit* { *circuit* is an Euler circuit}

53. If every vertex has even degree, then we can simply use Fleury's algorithm to find an Euler circuit, which is by definition also an Euler path. If there are two vertices with odd degree (and the rest with even degree), then we can add an edge between these two vertices and apply Fleury's algorithm (using this edge as the first edge to make it easier to find later), then delete the added edge.

55. A Hamilton circuit in a bipartite graph would have to look like $a_1, b_1, a_2, b_2, \ldots, a_k, b_k, a_1$, where each a_i is in one part and each b_i is in the other part, since the only edges in the graph join vertices in opposite parts. In the Hamilton circuit, no vertex is listed twice (except for the final a_1), and every vertex is listed, so the total number of vertices in the graph must be $2k$, which is not an odd number. Therefore a bipartite graph with an odd number of vertices cannot have a Hamilton circuit.

57. We draw one vertex for each of the 12 squares on the board. We then draw an edge from a vertex to each vertex that can be reached by moving 2 units horizontally and 1 unit vertically or vice versa. The result is as shown.



59. First let us try to find a reentrant knight's tour. Looking at the graph in the solution to Exercise 57 we see that every vertex on the left and right edge has degree 2. Therefore the 12 edges incident to these vertices would have to be in a Hamilton circuit if there were one. If we draw these 12 edges, however, we see that they form two circuits, each with six edges. Therefore there is no re-entrant knight's tour. However, we can splice these two circuits together by using an edge from a middle vertex in the top row to a middle vertex in the bottom row (and omitting two edges adjacent to this edge). The result is the knight's tour shown here.

3	6	9	12
8	11	4	1
5	2	7	10

61. We give an ad hoc argument by contradiction, using the notation shown in the following diagram. We think of the board as a graph and need to decide which edges need to be in a purported Hamilton path.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

There are only two moves from each of the four corner squares. If we put in all of the edges 1-10, 1-7, 16-10, and 16-7, then a circuit is complete too soon, so at least one of these edges must be missing. Without loss of generality, then, we may assume that the endpoints of the path are 1 and either 4 or 13, and that the path contains all of the edges 1-10, 10-16, and 16-7. Now vertex (square) 3 has edges only to squares 5, 10, and 12; and square 10 already has its two incident edges. Therefore 3-5 and 3-12 must be in the Hamilton path. Similarly, edges 8-2 and 8-15 must be in the path. Now square 9 has edges only to squares 2, 7, and 15. If there were to be edges to both 2 and 15, then a circuit would be completed too soon (2-9-15-8-2). Therefore the edge 9-7 must be in the path, thereby giving square 7 its full complement of edges. But now square 14 is forced to be joined in the path to squares 5 and 12, and this completes a circuit too soon (5-14-12-3-5). Since we have reached a contradiction, we conclude that there is no Hamilton path.

63. An $m \times n$ board contains mn squares. If both m and n are odd, then it contains an odd number of squares. By Exercise 62, the corresponding graph is bipartite. Exercise 55 told us that the graph does not contain a Hamilton circuit. Therefore there is no re-entrant knight's tour (see Exercise 58b).

65. This is a proof by contradiction. We assume that G satisfies Ore's inequality that $\deg(x) + \deg(y) \geq n$ whenever x and y are nonadjacent vertices in G, but G does not have a Hamilton circuit. We will end up with a contradiction, and therefore conclude that under these conditions, G must have a Hamilton circuit.

a) Since G does not have a Hamilton circuit, we can add missing edges one at a time in such a way that we do not obtain a graph with a Hamilton circuit. We continue this process as long as possible. Clearly it cannot go on forever, because once we've formed the complete graph by adding all missing edges, there is a Hamilton circuit (recall that $n \geq 3$). Whenever the process stops, we have obtained a graph H with the desired property. (Note that H might equal G itself—in other words, we add no edges. However, H cannot be complete, as just noted.)

b) Add one more edge to H. By the construction in part **(a)**, we now have a Hamilton circuit, and clearly this circuit must use the edge we just added. The path consisting of this circuit with the added edge omitted is clearly a Hamilton path in H.

c) Clearly v_1 and v_n are not adjacent in H, since H has no Hamilton circuit. Therefore they are not adjacent in G. But the hypothesis was that the sum of the degrees of vertices not adjacent in G was at least n. This inequality can be rewritten as $n - \deg(v_n) \leq \deg(v_1)$. But $n - \deg(v_n)$ is just the number of vertices not adjacent to v_n.

d) Let's make sure we understand what this means. If, say, v_7 is adjacent to v_1, then v_6 is in S. Note that $v_1 \in S$, since v_2 is adjacent to v_1. Also, v_n is not in S, since there is no vertex following v_n in the Hamilton path. Now each one of the $\deg(v_1)$ vertices adjacent to v_1 gives rise to an element of S, so S contains $\deg(v_1)$ vertices.

e) By part **(c)** there are at most $\deg(v_1) - 1$ vertices other than v_n not adjacent to v_n, and by part **(d)** there are $\deg(v_1)$ vertices in S, none of which is v_n. So S has more vertices other than v_n than there are vertices not adjacent to v_n; in other words, at least one vertex of S is adjacent to v_n. By definition, if v_k is

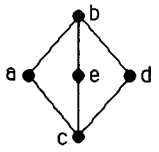
that $P(i)$ is maximized and following the C links.

SECTION 10.7 Planar Graphs

As with Euler and Hamilton circuits and paths, the topic of planar graphs is a classical one in graph theory. The theory (Euler's formula, Kuratowski's theorem, and their corollaries) is quite beautiful. It is easy to ask extremely difficult questions in this area, however—see Exercise 27, for example. In practice, there are very efficient algorithms for determining planarity that have nothing to do with Kuratowski's theorem, but they are quite complicated and beyond the scope of this book. For the exercises here, the best way to show that a graph is planar is to draw a planar embedding; the best way to show that a graph is nonplanar is to find a subgraph homeomorphic to K_5 or $K_{3,3}$. (Usually it will be $K_{3,3}$.)

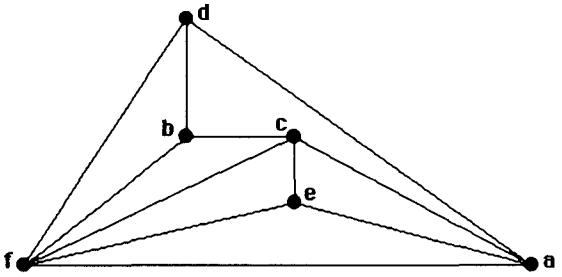
1. The question is whether $K_{5,2}$ is planar. It clearly is so, since we can draw it in the xy-plane by placing the five vertices in one part along the x-axis and the other two vertices on the positive and negative y-axis.

3. For convenience we label the vertices a, b, c, d, e, starting with the vertex in the lower left corner and proceeding clockwise around the outside of the figure as drawn in the exercise. This graph is just $K_{2,3}$; the picture below shows it redrawn by moving vertex c down.



5. This is $K_{3,3}$, with parts $\{a, d, f\}$ and $\{b, c, e\}$. Therefore it is not planar.

7. This graph can be untangled if we play with it long enough. The following picture gives a planar representation of it.



9. If one has access to software such as *The Geometer's Sketchpad*, then this problem can be solved by drawing the graph and moving the points around, trying to find a planar drawing. If we are unable to find one, then we look for a reason why—either a subgraph homeomorphic to K_5 or one homeomorphic to $K_{3,3}$ (always try the latter first). In this case we find that there is a homeomorphic copy of $K_{3,3}$, with vertices b, g, and i in one set and a, f, and h in the other; all the edges are there except for the edge bh, and it is represented by the path beh.

11. We give a proof by contradiction. Suppose that there is a planar representation of K_5, and let us call the vertices v_1, v_2, ..., v_5. There must be an edge from every vertex to every other. In particular, v_1, v_2, v_3, v_4, v_5, v_1 must form a pentagon. The pentagon separates the plane into two regions, an inside and an outside. The edge from v_1 to v_3 must be present, and without loss of generality let us assume it is drawn on the inside. Then there is no way for edges $\{v_2, v_4\}$ and $\{v_2, v_5\}$ to be in the inside, so they must be in the outside region. Now this prevents edges $\{v_1, v_4\}$ and $\{v_3, v_5\}$ from being on the outside. But they cannot both be on the inside without crossing. Therefore there is no planar representation of K_5.

13. We apply Euler's formula $r = e - v + 2$. Here we are told that $v = 6$. We are also told that each vertex has degree 4, so that the sum of the degrees is 24. Therefore by the handshaking theorem there are 12 edges, so $e = 12$. Solving, we find $r = 8$.

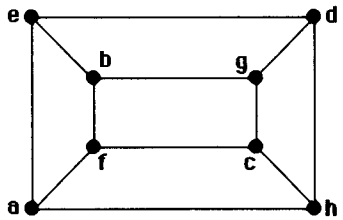
15. The proof is very similar to the proof of Corollary 1. First note that the degree of each region is at least 4. The reason for this is that there are no loops or multiple edges (which would give regions of degree 1 or 2) and no simple circuits of length 3 (which would give regions of degree 3); and the degree of the unbounded region is at least 4 since we are assuming that $v \geq 3$. Therefore we have, arguing as in the proof of Corollary 1, that $2e \geq 4r$, or simply $r \leq e/2$. Plugging this into Euler's formula, we obtain $e - v + 2 \leq e/2$, which gives $e \leq 2v - 4$ after some trivial algebra.

17. The proof is exactly the same as in Exercise 15, except that this time the degree of each region must be at least 5. Thus we get $2e \geq 5r$, which after the same algebra as before, gives the desired inequality.

19. **a)** If we remove a vertex from K_5, then we get K_4, which is clearly planar.
b) If we remove a vertex from K_6, then we get K_5, which is not planar.
c) If we remove a vertex from $K_{3,3}$, then we get $K_{3,2}$, which is clearly planar.
d) We assume the question means "Is it the case that for every v, the removal of v makes the graph planar?" Then the answer is no, since we can remove a vertex in the part of size 4 to leave $K_{3,3}$, which is not planar.

21. This graph is planar and hence cannot be homeomorphic to $K_{3,3}$.

23. The instructions are really not fair. It is hopeless to try to use Kuratowski's theorem to prove that a graph *is* planar, since we would have to check hundreds of cases to argue that there is no subgraph homeomorphic to K_5 or $K_{3,3}$. Thus we will show that this graph is planar simply by giving a planar representation. Note that it is Q_3.



25. This graph is nonplanar, since it contains $K_{3,3}$ as a subgraph: the parts are $\{a, g, d\}$ and $\{b, c, e\}$. (Actually it contains $K_{3,4}$, and it even contains a subgraph homeomorphic to K_5.)

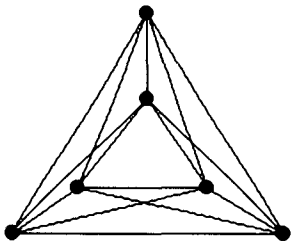
27. This is an extremely hard problem. We will present parts of the solution; the reader should consult a good graph theory book, such as Gary Chartrand, Linda Lesniak and Ping Zhang's *Graphs & Digraphs*, fifth edition (Chapman & Hall/CRC Press, 2011), for references and further details.

First we will state, without proof, what is known about crossing numbers for complete graphs (much is still not known about crossing numbers). If $n \leq 10$, then the crossing number of K_n is given by the following product

$$\frac{1}{4}\left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \left\lfloor \frac{n-2}{2} \right\rfloor \left\lfloor \frac{n-3}{2} \right\rfloor .$$

Thus the answers for parts **(a)**, **(b)**, and **(c)** are 1, 3, and 9, respectively. The figure below shows K_6 drawn in the plane with three crossings, which at least proves that the crossing number of K_6 is at most 3. The

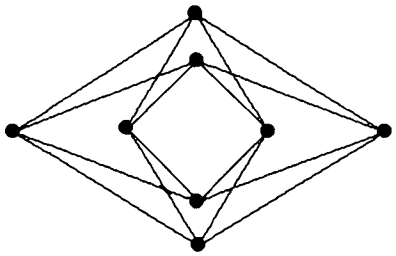
proof that it is not less than 3 is not easy. The embedding of K_5 with one crossing can be seen in this same picture, by ignoring the vertex at the top.



Second, for the complete bipartite graphs, what is known is that if the smaller of m and n is at most 6, then the crossing number of $K_{m,n}$ is given by the following product

$$\left\lfloor \frac{m}{2} \right\rfloor \left\lfloor \frac{m-1}{2} \right\rfloor \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor .$$

Thus the answers for parts **(d)**, **(e)**, and **(f)** are 2, 4, and 16, respectively. The figure below shows $K_{4,4}$ drawn in the plane with four crossings, which at least proves that the crossing number of $K_{4,4}$ is at most 4. The proof that it is not less than 4 is, again, difficult. It is also easy to see from this picture that the crossing number of $K_{3,4}$ is at most 2 (by ignoring the top vertex).



29. Let us follow the hint, and draw all the edges with straight line segments. This clearly produces a drawing of $K_{m,n}$. We will show that the number of crossings is $mn(m-2)(n-2)/16$, and that will complete the proof. (Incidentally, it is not known whether this upper bound is actually the crossing number. No one has found an embedding with fewer crossings, but only in the case in which the smaller of m and n is at most 6 has it been proved that it cannot be done. See the comments in the solution to Exercise 27.) In order to count the crossings, it is enough to count the crossings occurring in the first quadrant and multiply by 4. Let us label the points on the positive x-axis with the numbers 1 through $m/2$, and those on the y-axis with the numbers 1 through $n/2$. If we choose any two distinct numbers, say a and b with $a < b$, from 1 to $m/2$, and any two distinct numbers, say r and s with $r < s$, from 1 through $n/2$, then we get exactly one crossing in our graph, namely between the edges as and br. (There is no crossing between ar and bs.) So the number of crossings in the first quadrant is the same as the number of ways to make these choices, which is clearly $C(m/2,2) \cdot C(n/2,2)$. So the total number of crossings is 4 times this quantity, namely

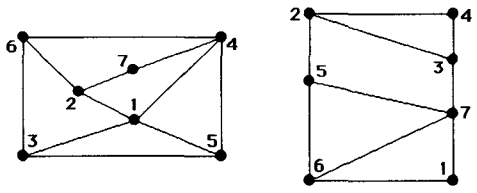
$$4 \cdot C(m/2,2) \cdot C(n/2,2) = 4 \cdot \frac{\frac{m}{2}\left(\frac{m}{2}-1\right)}{2} \cdot \frac{\frac{n}{2}\left(\frac{n}{2}-1\right)}{2},$$

which easily simplifies to

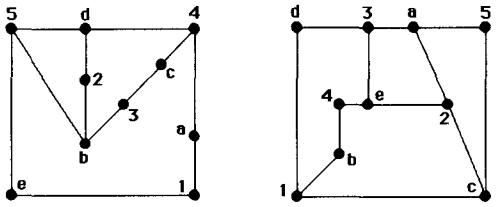
$$\frac{mn(m-2)(n-2)}{16}.$$

31. Each of these graphs is nonplanar; the first three contain K_5, and the last three contain $K_{3,3}$. Thus if we can show how to draw each of the graphs in two planes, then we will have shown that the thickness is 2 in each

case. The following picture shows that K_7 can be drawn in 2 planes, so this takes care of part **(a)**, part **(b)**, and part **(c)**.



The following picture shows that $K_{5,5}$ can be drawn in 2 planes, so this takes care of part **(d)**, part **(e)**, and part **(f)**.



33. The formula is certainly valid for $n \leq 4$, so let us assume that $n > 4$. By Exercise 32, the thickness of K_n is at least

$$\frac{C(n,2)}{3n-6} = \frac{n(n-1)/2}{3n-6} = \frac{n(n-1)}{6(n-2)} = \frac{1}{6}\left(n+1+\frac{2}{n-2}\right)$$

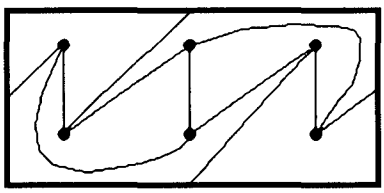
rounded up. Since this quantity is never an integer, it equals one more than itself rounded down, namely

$$\frac{1}{6}\left(n+1+\frac{2}{n-2}\right)+1 = \frac{n+7}{6}+\frac{2}{6(n-2)}$$

rounded down. The last term can be ignored: it is always less than $1/6$ and therefore will not influence the rounding process (since the first term has denominator 6). Thus we have proved that the thickness of K_n is at least $\lfloor (n+7)/6 \rfloor$.

35. This follows immediately from Exercise 34, since $K_{m,n}$ has mn edges and $m+n$ vertices and, being bipartite, has no triangles.

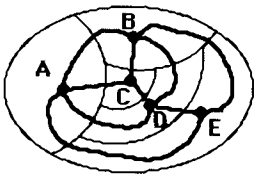
37. We can represent the surface of a torus with a rectangle, thinking of the right-hand edge as being equal to the left-hand edge, and the top edge as being equal to the bottom edge. For example, if we travel out of the rectangle across the right-hand edge about a third of the way from the top, then we immediately reenter the rectangle across the left-hand edge about a third of the way from the top. The picture below shows $K_{3,3}$ drawn on this surface. Note that the edges that seem to leave the rectangle really reenter it from the opposite side.



SECTION 10.8 Graph Coloring

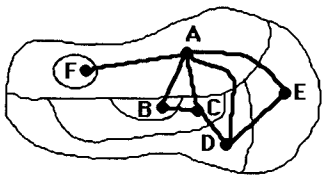
Like the problem of finding Hamilton paths, the problem of finding colorings with the fewest possible colors probably has no good algorithm for its solution. In working these exercises, for the most part you should proceed by trial and error, using whatever insight you can gain by staring at the graph (for instance, finding large complete subgraphs). There are also some interesting exercises here on coloring the edges of graphs—see Exercises 21–26. Exercises 29–31 are worth looking at, as well: they deal with a fast algorithm for coloring a graph that is not guaranteed to produce an optimal coloring.

1. We construct the dual graph by putting a vertex inside each region (but not in the unbounded region), and drawing an edge between two vertices if the regions share a common border. The easiest way to do this is illustrated in our answer. First we draw the map, then we put a vertex inside each region and make the connections. The dual graph, then, is the graph with heavy lines.



The number of colors needed to color this map is the same as the number of colors needed to color the dual graph. Since A, B, C, and D are mutually adjacent, at least four colors are needed. We can color each of the vertices (i.e., regions) A, B, C, and D a different color, and we can give E the same color as we give C.

3. We construct the dual as in Exercise 1.



As in Exercise 1, the number of colors needed to color this map is the same as the number of colors needed to color the dual graph. Three colors are clearly necessary, because of the triangle ABC, for instance. Furthermore three colors suffice, since we can color vertex (region) A red, vertices B, D, and F blue, and vertices C and E green.

5. For Exercises 5–11, in order to prove that the chromatic number is k, we need to find a k-coloring and to show that (at least) k colors are needed. Here, since there is a triangle, at least 3 colors are needed. Clearly 3 colors suffice, since we can color a and d the same color.

7. Since there is a triangle, at least 3 colors are needed. Clearly 3 colors suffice, since we can color a and c the same color.

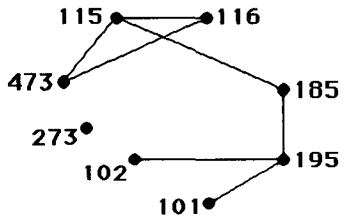
9. Since there is an edge, at least 2 colors are needed. The coloring in which b, d, and e are red and a and c blue shows that 2 colors suffice.

11. Since there is a triangle, at least 3 colors are needed. It is not hard to construct a 3-coloring. We can let a, f, h, j, and n be blue; let b, d, g, k, and m be green; and let c, e, i, l, and o be yellow.

13. If a graph has an edge (not a loop, since we are assuming that the graphs in this section are simple), then its chromatic number is at least 2. Conversely, if there are no edges, then the coloring in which every vertex receives the same color is proper. Therefore a graph has chromatic number 1 if and only if it has no edges.

15. In Example 4 we saw that the chromatic number of C_n is 2 if n is even and 3 if n is odd. Since the wheel W_n is just C_n with one more vertex, adjacent to all the vertices of the C_n along the rim of the wheel, W_n clearly needs exactly one more color than C_n (for that middle vertex). Therefore the chromatic number of W_n is 3 if n is even and 4 if n is odd.

17. Consider the graph representing this problem. The vertices are the 8 courses, and two courses are joined by an edge if there are students taking both of them. Thus there are edges between every pair of vertices except the 7 pairs listed. It is much easier to draw the complement than to draw this graph itself; it is shown below.



We want to find the chromatic number of the graph whose complement we have drawn; the colors will be the time periods for the exams. First note that since Math 185 and the four CS courses form a K_5 (in other words, there are no edges between any two of these in our picture), the chromatic number is at least 5. To show that it equals 5, we just need to color the other three vertices. A little trial and error shows that we can make Math 195 the same color as (i.e., have its final exam at the same time as) CS 101; and we can make Math 115 and 116 the same color as CS 473. Therefore five time slots (colors) are sufficient.

19. We model the problem with the intersection graph of these sets. Note that every pair of these intersect except for C_4 and C_5. Thus the graph is K_6 with that one edge deleted. Clearly its chromatic number is 5, since we need to color all the vertices different colors, except that C_4 and C_5 may have the same color. In other words, 5 meeting times are needed, since only committees C_4 and C_5 can meet simultaneously.

21. Note that the number of colors needed to color the edges is at least as large as the largest degree of a vertex, since the edges at each vertex must all be colored differently. Hence if we can find an edge coloring with that many colors, then we know we have found the answer. In Exercise 5 there is a vertex of degree 3, so the edge chromatic number is at least 3. On the other hand, we can color $\{a, c\}$ and $\{b, d\}$ the same color, so 3 colors suffice. In Exercise 6 the 6 edges incident to g must all get different colors. On the other hand, it is not hard to complete a proper edge coloring with only these colors (for example, color edge $\{a, f\}$ with the same color as used on $\{d, g\}$), so the answer is 6. In Exercise 7 the answer must be at least 3; it is 3 since edges that appear as parallel line segments in the picture can have the same color. In Exercise 8 clearly 4 colors are required, since the vertices have degree 4. In fact 4 colors are sufficient. Here is one proper 4-coloring (we denote edges in the obvious shorthand notation): color 1 for ac, be, and df; color 2 for ae, bd, and cf; color 3 for ab, cd, and ef; and color 4 for ad, bf, and ce. In Exercise 9 the answer must be at least 3; it is easy to construct a 3-coloring of the edges by inspection: $\{a, b\}$ and $\{c, e\}$ have the same color, $\{a, d\}$ and $\{b, c\}$ have the same color, and $\{a, e\}$ and $\{c, d\}$ have the same color. In Exercise 10 the largest degree is 6 (vertex i has degree 6); therefore at least 6 colors are required. By trial and error we come up with this coloring using 6 colors (we use the obvious shorthand notation for edges); there are many others, of course. Assign color 1 to ag, cd, and hi; color 2 to ab, cf, dg, and ei; color 3 to bh, cg, di, and ef; color 4 to ah, ci, and de; color 5 to bi, ch, and fg; and color 6 to ai, bc, and gh. Finally, in Exercise 11 it is easy to construct an edge-coloring with 4 colors; again the edge chromatic number is the maximum degree of a vertex.

Despite the appearances of these examples, it is not the case that the edge chromatic number of a graph is always equal to the maximum degree of the vertices in the graph. The simplest example in which this is not

true is K_3. Clearly its edge chromatic number is 3 (since all three edges are adjacent to each other), but its maximum degree is 2. There is a theorem, however, stating that the edge chromatic number is always equal to either the maximum degree or one more than the maximum degree.

23. **a)** The n-cycle's edges are just like the n-cycle's vertices (each adjacent to the next as we go around the cycle), so the edge chromatic number is the same, namely 2 if n is even and 3 if n is odd, as in Example 4. **b)** The edge chromatic number is at least n, because the radial edges are all pairwise adjacent and therefore must all have distinct colors. Suppose we call these colors 1 through n proceeding clockwise. We need no additional colors for the edges of the cycle, because we can color the edge adjacent to the spokes colored 1 and 2 with color 3 and proceed clockwise with colors 4, 5, ..., $n-1$, n, 1, and 2. Therefore $\chi'(W_n) = n$.

25. Two edges that have the same color share no endpoints. Therefore if more than $n/2$ edges were colored the same, the graph would have more than $2(n/2) = n$ vertices.

27. This problem can be modeled with the intersection graph of the sets of steps during which the variables must be stored. This graph has 7 vertices, t through z; there is an edge between two vertices if the two variables they represent must be stored during some common step. The answer to the problem is the chromatic number of this graph. Rather than considering this graph, we look at its complement (it has a lot fewer edges). Here two vertices are adjacent if the sets (of steps) do not intersect. The only edges are $\{u,w\}$, $\{u,x\}$, $\{u,y\}$, $\{u,z\}$, $\{v,x\}$, $\{x,z\}$. Note that there are no edges in the complement joining any two of $\{t,v,w,y,z\}$, so that these vertices form a K_5 in the original graph. Thus the chromatic number of the original graph is at least 5. To see that it is 5, note that vertex u can have the same color as w, and x can have the same color as z (these pairs appear as edges in the complement). Since the chromatic number is 5, we need 5 registers, with variables u and w sharing a register, and vertices x and z sharing one.

29. First we need to list the vertices in decreasing order of degree. This ordering is not unique, of course; we will pick e,a,b,c,f,h,i,d,g,j. Next we assign color 1 to e, and then to f and d, in that order. Now we assign color 2 to a, c, i, and g, in that order. Finally, we assign color 3 to b, h and j, in that order. Thus the algorithm gives a 3-coloring. Since the graph contains triangles, we know that this is the best possible, so the algorithm "worked" here (but it need not always work—see Exercise 27).

31. A simple example in which the algorithm may fail to provide a coloring with the minimum number of colors is C_6, which of course has chromatic number 2. Since all the vertices are of degree 2, we may order them v_1, v_4, v_2, v_3, v_5, v_6, where the edges are $\{v_1,v_2\}$, $\{v_2,v_3\}$, $\{v_3,v_4\}$, $\{v_4,v_5\}$, $\{v_5,v_6\}$, and $\{v_1,v_6\}$. Then v_1 gets color 1, as does v_4. Next v_2 and v_5 get color 2; and then v_3 and v_6 must get color 3.

33. We need to show that the wheel W_n when n is an odd integer greater than 1 can be colored with four colors, but that any graph obtained from it by removing one edge can be colored with three colors. Four colors are needed to color this graph, because three colors are needed for the rim (see Example 4), and the center vertex, being adjacent to all the rim vertices, will require a fourth color. To complete the proof that W_n is chromatically 4-critical, we must show that the graph obtained from W_n by deleting one edge can be colored with three colors. There are two cases. If we remove a rim edge, then we can color the rim with two colors, by starting at an endpoint of the removed edge and using the colors alternately around the portion of the rim that remains. The third color is then assigned to the center vertex. On the other hand, if we remove a spoke edge, then we can color the rim by assigning color #1 to the rim endpoint of the removed edge and colors #2 and #3 alternately to the remaining vertices on the rim, and then assign color #1 to the center.

35. We give a proof by contradiction. Suppose that G is chromatically k-critical but has a vertex v of degree $k-2$ or less. Remove from G one of the edges incident to v. By definition of "k-critical," the resulting graph can be colored with $k-1$ colors. Now restore the missing edge and use this coloring for all vertices except v. Because we had a proper coloring of the smaller graph, no two adjacent vertices have the same color. Furthermore, v has at most $k-2$ neighbors, so we can color v with an unused color to obtain a proper $(k-1)$-coloring of G. This contradicts the fact that G has chromatic number k. Therefore our assumption was wrong, and every vertex of G must have degree at least $k-1$.

37. **a)** Note that vertices d, e, and f are mutually adjacent. Therefore six different colors are needed in a 2-tuple coloring, since each of these three vertices needs a disjoint set of two colors. In fact it is easy to give a coloring with just six colors: Color a, d, and g with $\{1,2\}$; color c and e with $\{3,4\}$; and color b and f with $\{5,6\}$. Thus $\chi_2(G) = 6$.

b) This one is trickier than part **(a)**. There is no coloring with just six colors, since if there were, we would be forced (without loss of generality) to color d with $\{1,2\}$; e with $\{3,4\}$; f with $\{5,6\}$; then g with $\{1,2\}$, b with $\{5,6\}$, and c with $\{3,4\}$. This gives no free colors for vertex a. Now this may make it appear that eight colors are required, but a little trial and error shows us that seven suffice: Color a with $\{2,4\}$; color b and f with $\{5,6\}$; color d with $\{1,2\}$; color c with $\{3,7\}$; color e with $\{3,4\}$; and color g with $\{1,7\}$. Thus $\chi_2(H) = 7$.

c) This is similar to part **(a)**. Here nine colors are necessary and sufficient, since a, d, and g can get one set of three colors; b and f can get a second set; and c and e can get a third set. Clearly nine colors are necessary to color the triangles.

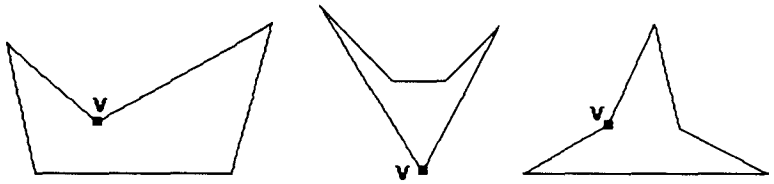
d) First we construct a coloring with 11 colors: Color a with $\{3,6,11\}$; color b and f with $\{7,8,9\}$; color d with $\{1,2,3\}$; color c with $\{4,5,10\}$; color e with $\{4,6,11\}$; and color g with $\{1,2,5\}$. To prove that $\chi_3(H) = 11$, we must show that it is impossible to give a 3-tuple coloring with only ten colors. If such a coloring were possible, without loss of generality we could color d with $\{1,2,3\}$, e with $\{4,5,6\}$, f with $\{7,8,9\}$, and g with $\{1,2,10\}$. Now nine colors are needed for the three vertices a, b, and c, since they form a triangle; but colors 1 and 2 are already used in vertices adjacent to all three of them. Therefore at least $9+2 = 11$ colors are necessary.

39. The frequencies are the colors, the zones are the vertices, and two zones that are so close that interference would be a problem are joined by an edge in the graph. Then it is clear that a k-tuple coloring is exactly an assignment of frequencies that avoids possible interference.

41. We use induction on the number of vertices of the graph. Every graph with five or fewer vertices can be colored with five or fewer colors, since each vertex can get a different color. That takes care of the basis case(s). So we assume that all graphs with k vertices can be 5-colored and consider a graph G with $k+1$ vertices. By Corollary 2 in Section 10.7, G has a vertex v with degree at most 5. Remove v to form the graph G'. Since G' has only k vertices, we 5-color it by the inductive hypothesis. If the neighbors of v do not use all five colors, then we can 5-color G by assigning to v a color not used by any of its neighbors. The difficulty arises if v has five neighbors, and each has a different color in the 5-coloring of G'. Suppose that the neighbors of v, when considered in clockwise order around v, are a, b, c, m, and p. (This order is determined by the clockwise order of the curves representing the edges incident to v.) Suppose that the colors of the neighbors are azure, blue, chartreuse, magenta, and purple, respectively. Consider the azure-chartreuse subgraph (i.e., the vertices in G colored azure or chartreuse and all the edges between them). If a and c are not in the same component of this graph, then in the component containing a we can interchange these two colors (make the azure vertices chartreuse and vice versa), and G' will still be properly colored. That makes a chartreuse, so we can now color v azure, and G has been properly colored. If a and c are in the same component, then there is a path of vertices alternately colored azure and chartreuse joining a and c. This path together with

edges av and vc divides the plane into two regions, with b in one of them and m in the other. If we now interchange blue and magenta on all the vertices in the same region as b, we will still have a proper coloring of G', but now blue is available for v. In this case, too, we have found a proper coloring of G. This completes the inductive step, and the theorem is proved.

43. We follow the hint. Because the measures of the interior angles of a pentagon total $540°$, there cannot be as many as three interior angles of measure more than $180°$ (reflex angles). If there are no reflex angles, then the pentagon is convex, and a guard placed at any vertex can see all points. If there is one reflex angle, then the pentagon must look essentially like figure (a) below, and a guard at vertex v can see all points. If there are two reflex angles, then they can be adjacent or nonadjacent (figures (b) and (c)); in either case, a guard at vertex v can see all points. (In figure (c), choose the reflex vertex closer to the bottom side.) Thus for all pentagons, one guard suffices, so $g(5) = 1$.



45. The figure suggested in the hint (generalized to have k prongs for any $k \geq 1$) has $3k$ vertices. Consider the set of points from which a guard can see the tip of the first prong, the set of points from which a guard can see the tip of the second prong, and so on. These are disjoint triangles (together with their interiors). Therefore a separate guard is needed for each of the k prongs, so at least k guards are needed. This shows that $g(3k) \geq k = \lfloor 3k/3 \rfloor$. To handle values of n that are not multiples of 3, let $n = 3k + i$, where $i = 1$ or 2. Then obviously $g(n) \geq g(3k) \geq k = \lfloor n/3 \rfloor$.

GUIDE TO REVIEW QUESTIONS FOR CHAPTER 10

1. **a)** See pp. 641–642 and Table 1 in Section 10. **b)** See Exercise 1 in Section 10.1.

2. See all the examples Section 10.1.

3. See Theorem 1 in Section 10.2.

4. See Theorem 2 in Section 10.2.

5. See Theorem 3 in Section 10.2.

6. **a)** See Example 5 in Section 10.2. **b)** See Example 13 in Section 10.2.
c) See Example 6 in Section 10.2. **d)** See Example 7 in Section 10.2.
e) See Example 8 in Section 10.2.

7. **a)** n, $C(n,2)$ **b)** $m+n$, mn **c)** n, n **d)** $n+1$, $2n$ **e)** 2^n, $n2^{n-1}$

8. **a)** See p. 656. **b)** K_2 and C_{2m}
c) (See also Example 12 and Exercise 66 in Section 10.2.) The following algorithm is an efficient way to determine whether a connected graph can be 2-colored (which is the same thing as saying that it is bipartite); apply it to each component of the given graph. First color any vertex red. Then color all vertices adjacent to this vertex blue. Then look at all vertices adjacent to these just-colored blue vertices. If any of them are already colored blue, then stop and declare the graph not to be bipartite; otherwise color all the uncolored ones red. Next look at all vertices adjacent to all the vertices just colored red. If any of them are already

CHAPTER 11
Trees

SECTION 11.1 Introduction to Trees

These exercises give the reader experience working with tree terminology, and in particular with the relationships between the height and the numbers of vertices, leaves, and internal vertices of a tree. Exercise 13 should be done to get a feeling for the structure of trees. One good way to organize your enumeration of trees (such as all nonisomorphic trees with five vertices) is to focus on a particular parameter, such as the length of a longest path in the tree. This makes it easier to include all the trees and not count any of them twice. Review the theorems in this section before working the exercises involving the relationships between the height and the numbers of vertices, leaves, and internal vertices of a tree. For a challenge that gives a good feeling for the flavor of arguments in graph theory, the reader should try Exercise 43. In many ways trees are recursive creatures, and Exercises 45 and 46 are worth looking at in this regard.

1. **a)** This graph is connected and has no simple circuits, so it is a tree.

b) This graph is not connected, so it is not a tree.

c) This graph is connected and has no simple circuits, so it is a tree.

d) This graph has a simple circuit, so it is not a tree.

e) This graph is connected and has no simple circuits, so it is a tree.

f) This graph has a simple circuit, so it is not a tree.

3. **a)** Vertex a is the root, since it is drawn at the top.

b) The internal vertices are the vertices with children, namely a, b, c, d, f, h, j, q, and t.

c) The leaves are the vertices without children, namely e, g, i, k, l, m, n, o, p, r, s, and u.

d) The children of j are the vertices adjacent to j and below j, namely q and r.

e) The parent of h is the vertex adjacent to h and above h, namely c.

f) Vertex o has only one sibling, namely p, which is the other child of o's parent, h.

g) The ancestors of m are all the vertices on the unique simple path from m back to the root, namely f, b, and a.

h) The descendants of b are all the vertices that have b as an ancestor, namely e, f, l, m, and n.

5. This is not a full m-ary tree for any m. It is an m-ary tree for all $m \geq 3$, since each vertex has at most 3 children, but since some vertices have 3 children, while others have 1 or 2, it is not full for any m.

7. We can easily determine the levels from the drawing. The root a is at level 0. The vertices in the row below a are at level 1, namely b, c, and d. The vertices below that, namely e through k (in alphabetical order), are at level 2. Similarly l through r are at level 3, s and t are at level 4, and u is at level 5.

9. We describe the answers, rather than actually drawing pictures.

a) The subtree rooted at a is the entire tree, since a is the root.

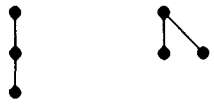
b) The subtree rooted at c consists of five vertices—the root c, children g and h of this root, and grandchildren o and p—and the four edges cg, ch, ho, and hp.

c) The subtree rooted at e is just the vertex e.

11. We find the answer by carefully enumerating these trees, i.e., drawing a full set of nonisomorphic trees. One way to organize this work so as to avoid leaving any trees out or counting the same tree (up to isomorphism) more than once is to list the trees by the length of their longest simple path (or longest simple path from the root in the case of rooted trees).

a) There is only one tree with three vertices, namely $K_{1,2}$ (which can also be thought of as the simple path of length 2).

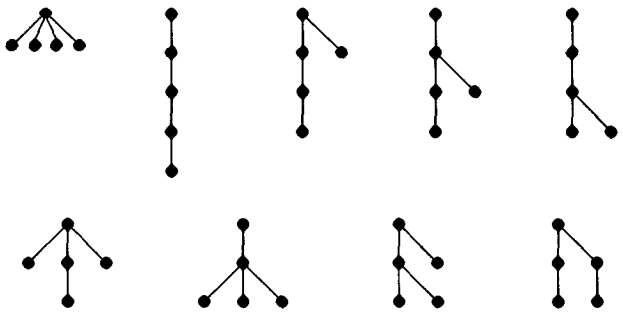
b) With three vertices, the longest path from the root can have length 1 or 2. There is only one tree of each type, so there are exactly two nonisomorphic rooted trees with 3 vertices, as shown below.



13. We find the answer by carefully enumerating these trees, i.e., drawing a full set of nonisomorphic trees. One way to organize this work so as to avoid leaving any trees out or counting the same tree (up to isomorphism) more than once is to list the trees by the length of their longest simple path (or longest simple path from the root in the case of rooted trees).

a) If the longest simple path has length 4, then the entire tree is just this path. If the longest simple path has length 3, then the fifth vertex must be attached to one of the middle vertices of this path. If the longest simple path has length 2, then the tree is just $K_{1,4}$. Thus there are only three trees with five vertices. They can be pictured as the first, second, and fourth pictures in the top row below.

b) For rooted trees of length 5, the longest path from the root can have length 1, 2, 3 or 4. There is only one tree with longest path of length 1 (the other four vertices are at level 1), and only one with longest path of length 4. If the longest path has length 3, then the fifth vertex (after using four vertices to draw this path) can be "attached" to either the root or the vertex at level 1 or the vertex at level 2, giving us three nonisomorphic trees. If the longest path has length 2, then there are several possibilities for where the fourth and fifth vertices can be "attached." They can both be adjacent to the root; they can both be adjacent to the vertex at level 1; one can be adjacent to the root and the other to the vertex at level 1; or one can be adjacent to the root and the other to this vertex: in all there are four possibilities in this case. Thus there are a total of nine nonisomorphic rooted trees on 5 vertices, as shown below.



15. a) We will prove this statement using mathematical induction on n, the number of vertices of G. (This exercise can also be done by using Exercise 14 and Theorem 2. Such a proof is given in the answer section of the textbook.) If $n = 1$, then there is only one possibility for G, it *is* a tree, it is connected, and it has $1 - 1 = 0$ edges. Thus the statement is true. Now let us assume that the statement is true for simple graphs with n vertices, and let G be a simple graph with $n + 1$ vertices.

There are two things to prove here. First let us suppose that G is a tree; we must show that G is connected and has $(n + 1) - 1 = n$ edges. Of course G is connected by definition. In order to prove that G

has the required number of edges, we need the following fact: a tree with at least one edge must contain a vertex of degree 1. (To see that this is so, let P be a simple path of greatest possible length; since the tree has no simple circuits, such a maximum length simple path exists. The ends of this path must be vertices of degree 1, since otherwise the simple path could be extended.) Let v be a vertex of degree 1 in G, and let G' be G with v and its incident edge removed. Now G' is still a tree: it has no simple circuits (since G had none) and it is still connected (the removed edge is clearly not needed to form paths between vertices different from v). Therefore by the inductive hypothesis, G', which has n vertices, has $n-1$ edges; it follows that G, which has one more edge than G', has n edges.

Conversely, suppose that G is connected and has n edges. If G is not a tree, then it must contain a simple circuit. If we remove one edge from this simple circuit, then the resulting graph (call it G') is still connected. If G' is a tree then we stop; otherwise we repeat this process. Since G had only finitely many edges to begin with, this process must eventually terminate at some tree T with $n+1$ vertices (T has all the vertices that G had). By the paragraph above, T therefore has n edges. But this contradicts the fact that we removed at least one edge of G in order to construct T. Therefore our assumption that G was not a tree is wrong, and our proof is complete.

b) For the "only if" direction, suppose that G is a tree. By part **(a)**, G has $n-1$ edges, and by definition, G has no simple circuits. For the "if" direction, suppose that G has no simple circuits and has $n-1$ edges. The only thing left to prove is that G is connected. Let c equal the number of components of G, each of which is necessarily a tree, say with n_i vertices, where $\sum_{i=1}^{c} n_i = n$. By part **(a)**, the total number of edges in G is $\sum_{i=1}^{c}(n_i - 1) = n - c$. Since we are given that this equals $n-1$, it follows that $c = 1$, i.e., G is connected.

17. Since a tree with n vertices has $n-1$ edges, the answer is 9999.

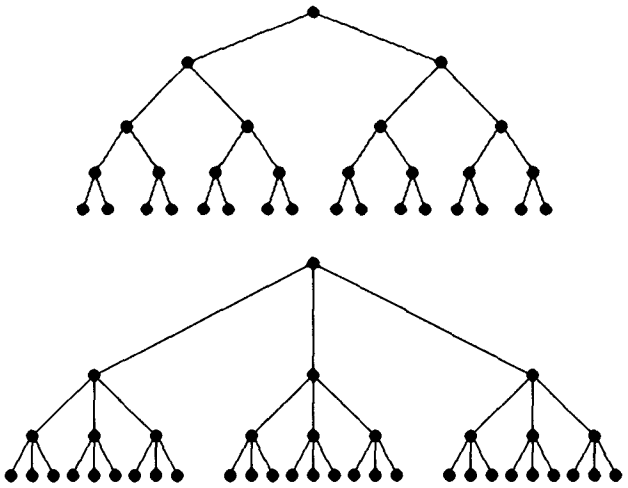
19. Each internal vertex has exactly 2 edges leading from it to its children. Therefore we can count the edges by multiplying the number of internal vertices by 2. Thus there are $2 \cdot 1000 = 2000$ edges.

21. We can model the tournament as a full binary tree. Each internal vertex represents the winner of the game played by its two children. There are 1000 leaves, one for each contestant. The root is the winner of the entire tournament. By Theorem 4(*iii*), with $m = 2$ and $l = 1000$, we see that $i = (l-1)/(m-1) = 999$. Thus exactly 999 games must be played to determine the champion.

23. Let P be a person sending out the letter. Then 10 people receive a letter with P's name at the bottom of the list (in the sixth position). Later 100 people receive a letter with P's name in the fifth position. Similarly, 1000 people receive a letter with P's name in the fourth position, and so on, until 1,000,000 people receive the letter with P's name in the first position. Therefore P should receive \$1,000,000. The model here is a full 10-ary tree.

25. No such tree exists. Suppose it did. By Theorem 4(*iii*), we know that a tree with these parameters must have $i = 83/(m-1)$ internal vertices. In order for this to be a whole number, $m-1$ must be a divisor of 83. Since 83 is prime, this means that $m = 2$ or $m = 84$. If $m = 2$, then we can have at most 15 vertices in all (the root, two at level 1, four at level 2, and eight at level 3). So m cannot be 2. If $m = 84$, then $i = 1$, which tells us that the root is the only internal vertex, and hence the height is only 1, rather than the desired 3. These contradictions tell us that no tree with 84 leaves and height 3 exists.

27. The complete binary tree of height 4 has 5 rows of vertices (levels 0 through 4), with each vertex not in the bottom row having two children. The complete 3-ary tree of height 3 has 4 rows of vertices (levels 0 through 3), with each vertex not in the bottom row having three children.



29. For both parts we use algebra on the equations $n = i + l$ (which is true by definition) and $n = mi + 1$ (which is proved in Theorem 3).

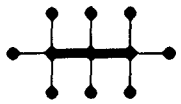
a) That $n = mi + 1$ is one of the given equations. For the second equality here, we have $l = n - i = (mi + 1) - i = (m - 1)i + 1$.

b) If we subtract the two given equations, then we obtain $0 = (1-m)i + (l-1)$, or $(m-1)i = l-1$. It follows that $i = (l-1)/(m-1)$. Then $n = i+l = [(l-1)/(m-1)]+l = (l-1+lm-l)/(m-1) = (lm-1)/(m-1)$.

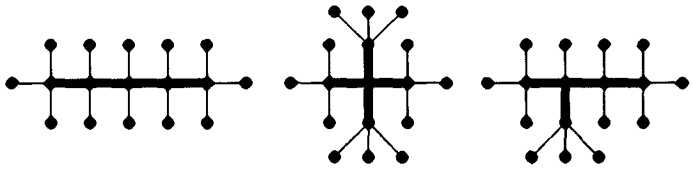
31. In each of the t trees, there is one fewer edge than there are vertices. Therefore altogether there are t fewer edges than vertices. Thus there are $n - t$ edges.

33. The number of isomers is the number of nonisomorphic trees with the given numbers of atoms. Since the hydrogen atoms play no role in determining the structure (they simply are attached to each carbon atom in sufficient number to make the degree of each carbon atom exactly 4), we need only look at the trees formed by the carbon atoms. In drawing our answers, we will show the tree of carbon atoms in heavy lines, with the hydrogen atom attachments in thinner lines.

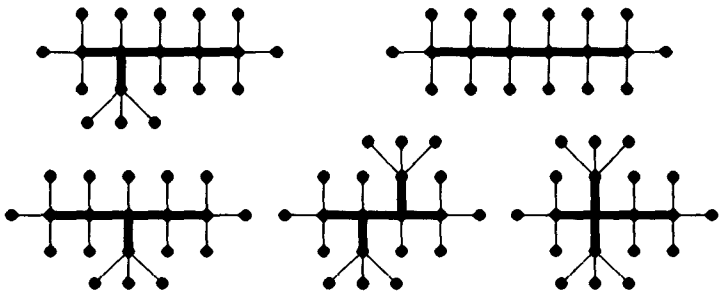
a) There is only one tree with three vertices (up to isomorphism), the path of length 2. Thus the answer is 1. The heavy lines in this diagram of the molecule form this tree.



b) There are 3 nonisomorphic trees with 5 vertices: the path of length 4, the "star" $K_{1,4}$, and the tree that consists of a path of length 3 together with one more vertex attached to one of the middle vertices in the path. Thus the answer is 3. Again the heavy lines in the diagrams of the molecules form these trees.



c) We need to find all the nonisomorphic trees with 6 vertices, except that we must not count the (one) tree with a vertex of degree 5 (since each carbon can only be attached to four other atoms). The complete set of trees is shown below (the heavy lines in these diagrams). Thus the answer is 5.



35. **a)** The parent of a vertex v is the directory in which the file or directory represented by v is contained.

b) The child of a vertex v (and v must represent a directory) is a file or directory contained in the directory that v represents.

c) If u and v are siblings, then the files or directories that u and v represent are in the same directory.

d) The ancestors of vertex v are all directories in the path from the root directory to the file or directory represented by v.

e) The descendants of a vertex v are all the files and directories either contained in v, or contained in directories contained in v, etc.

f) The level of a vertex v tells how far from the root directory is the file or directory represented by v.

g) The height of the tree is the greatest depth (i.e., level) at which a file or directory is buried in the system.

37. Suppose that $n = 2^k$, where k is a positive integer. We want to show how to add n numbers in $\log n$ steps using a tree-connected network of $n-1$ processors (recall that $\log n$ means $\log_2 n$). Let us prove this by mathematical induction on k. If $k = 1$ there is nothing to prove, since then $n = 2$ and $n - 1 = 1$, and certainly in $\log 2 = 1$ step we can add 2 numbers with 1 processor. Assume the inductive hypothesis, that we can add $n = 2^k$ numbers in $\log n$ steps using a tree-connected network of $n-1$ processors. Suppose now that we have $2n = 2^{k+1}$ numbers to add, $x_1, x_2, \ldots, x_{2n}$. The tree-connected network of $2n-1$ processors consists of the tree-connected network of $n-1$ processors together with two new processors as children of each leaf in the $(n-1)$-processor network. In one step we can use the leaves of the larger network to add $x_1 + x_2$, $x_3 + x_4$, $\ldots$, $x_{2n-1} + x_{2n}$. This gives us n numbers. By the inductive hypothesis we can now use the rest of the network to add these numbers using $\log n$ steps. In all, then, we used $1 + (\log n)$ steps, and, just as desired, $\log(2n) = \log 2 + \log n = 1 + \log n$. This completes the proof.

39. We need to compute the eccentricity of each vertex in order to find the center or centers. In practice, this does not involve much computation, since we can tell at a glance when the eccentricity is large. Intuitively, the center or centers are near the "middle" of the tree. The eccentricity of vertex c is 3, and it is the only vertex with eccentricity this small. Indeed, vertices a and b have eccentricities 4 and 5 (look at the paths to l); vertices d, f, g, j, and k all have eccentricities at least 4 (again look at the paths to l); and vertices e, h, i, and l also all have eccentricities at least 4 (look at the paths to k). Therefore c is the only center.

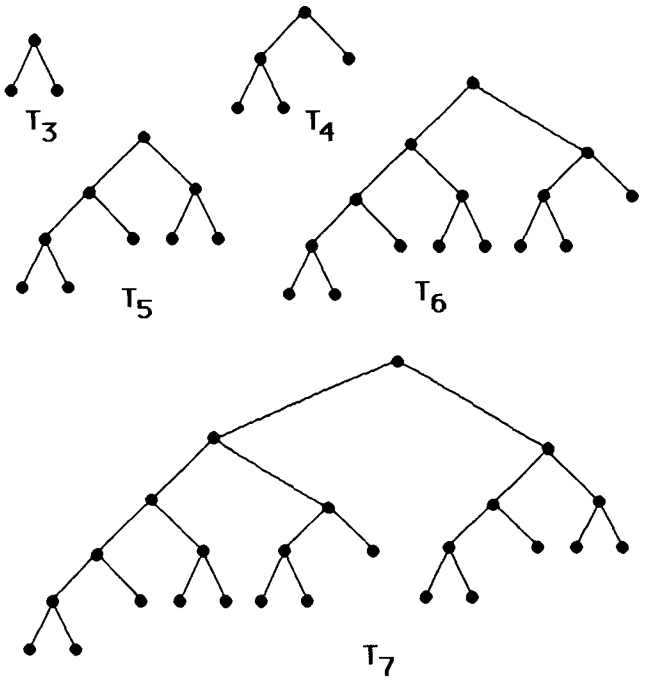
41. See the comments for the solution to Exercise 39. The eccentricity of vertices c and h are both 3. The eccentricities of the other vertices are all at least 4. Therefore c and h are the centers.

43. Certainly a tree has at least one center, since the set of eccentricities has a minimum value. First we prove that if u and v are any two distinct centers (say with minimum eccentricity e), then u and v are adjacent. Let P be the unique simple path from u to v. We will show that P is just u, v. If not, let c be any other vertex on P. Since the eccentricity of c is at least e, there is a vertex w such that the unique simple path Q from c to w has length at least e. This path Q may follow P for awhile, but once it diverges from P it cannot rejoin P without there being a simple circuit in the tree. In any case, Q cannot follow P towards

both u and v, so suppose without loss of generality that it does not follow P towards u. Then the path from u to c and then on to w is simple and of length greater than e, a contradiction. Thus no such c exists, and u and v are adjacent.

Finally, to see that there can be no more than two centers, note that we have just proved that every two centers are adjacent. If there were three (or more) centers, then we would have a K_3 contained in the tree, contradicting the definition that a tree has no simple circuits.

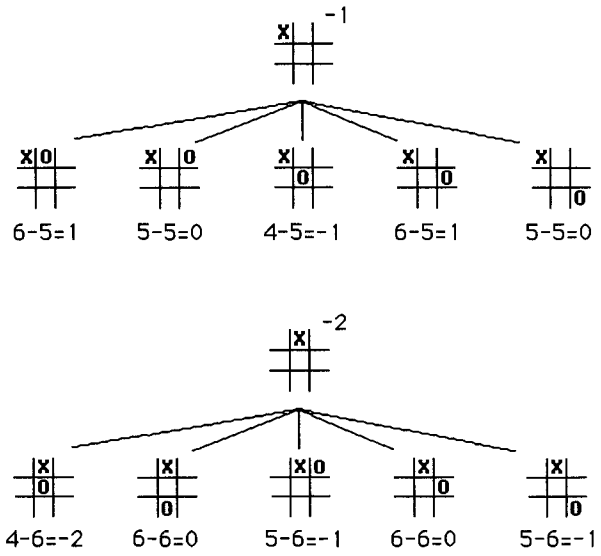
45. We follow the recursive definition and produce the following pictures for T_3 through T_7 (of course T_1 and T_2 are both the tree with just one vertex). For example, T_3 has T_2 (a single vertex) as its left subtree and T_1 (again a single vertex) as its right subtree.



47. This "proof" shows that *there exists* a tree with n vertices having a path of length $n-1$. Note that the inductive step correctly takes the tree whose existence is guaranteed by the inductive hypothesis and correctly constructs a tree of the desired type. However, the statement was that *every* tree with n vertices has a path of length $n-1$, and this was not shown. A proof of the inductive step would need to start with an arbitrary tree with $n+1$ vertices and show that it had the required path. Of course no such proof is possible, since the statement is not true. Douglas West, whose *Introduction to Graph Theory* is an excellent book on that subject, calls this mistake the induction trap.

SECTION 11.2 Applications of Trees

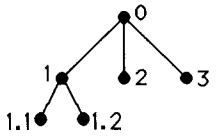
Trees find many applications, especially in computer science. This section and subsequent ones deal with some of these applications. Binary search trees can be built up by adding new vertices one by one; searches in binary search trees are accomplished by moving down the tree until the desired vertex is found, branching either right or left as necessary. Huffman codes provide efficient means of encoding text in which some symbols occur more frequently than others; decoding is accomplished by moving down a binary tree. The coin-weighing problems presented here are but a few of the questions that can be asked. Try making up some of your own and answering them; it is easy to ask quite difficult questions of this type.



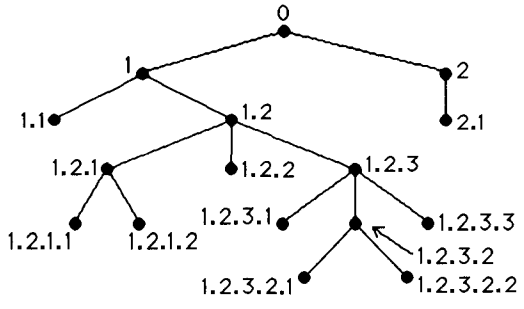
SECTION 11.3 Tree Traversal

Tree traversal is central to computer science applications. Trees are such a natural way to represent arithmetical and algebraic formulae, and so easy to manipulate, that it would be difficult to imagine how computer scientists could live without them. To see if you really understand the various orders, try Exercises 26 and 27. You need to make your mind work recursively for tree traversals: when you come to a subtree, you need to remember where to continue after processing the subtree. It is best to think of these traversals in terms of the recursive algorithms (shown as Algorithms 1, 2, and 3). A good bench-mark for testing your understanding of recursive definitions is provided in Exercises 30–34.

1. The root of the tree is labeled 0. The children of the root are labeled 1, 2, ..., from left to right. The children of a vertex labeled α are labeled $\alpha.1$, $\alpha.2$, ..., from left to right. For example, the two children of the vertex 1 here are 1.1 and 1.2. We completely label the tree in this manner, from the top down. See the figure. The lexicographic order of the labels is the preorder of the vertices: after each vertex come the subtrees rooted at its children, from left to right. Thus the order is $0 < 1 < 1.1 < 1.2 < 2 < 3$.



3. See the comments for the solution to Exercise 1. The order is $0 < 1 < 1.1 < 1.2 < 1.2.1 < 1.2.1.1 < 1.2.1.2 < 1.2.2 < 1.2.3 < 1.2.3.1 < 1.2.3.2 < 1.2.3.2.1 < 1.2.3.2.2 < 1.2.3.3 < 2 < 2.1$.



5. The given information tells us that the root has two children. We have no way to tell how many vertices are in the subtree of the root rooted at the first of these children. Therefore we have no way to tell how many vertices are in the tree.

7. In preorder, the root comes first, then the left subtree in preorder, then the right subtree in preorder. Thus the preorder is a, followed by the vertices of the left subtree (the one rooted at b) in preorder, then c. Recursively, the preorder in the subtree rooted at b is b, followed by d, followed by the vertices in the subtree rooted at e in preorder, namely e, f, g. Putting this all together, we obtain the answer a, b, d, e, f, g, c.

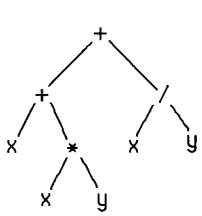
9. See the comments in the solution to Exercise 7 for the procedure. The only difference here is that some vertices have more than two children: after listing such a vertex, we list the vertices of its subtrees, in preorder, from left to right. The answer is $a, b, e, k, l, m, f, g, n, r, s, c, d, h, o, i, j, p, q$.

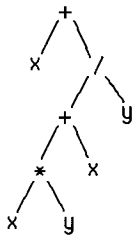
11. Inorder traversal requires that the left-most subtree be traversed first, then the root, then the remaining subtrees (if any) from left to right. Applying this principle, we see that the list must start with the left subtree in inorder. To find this, we need to start with *its* left subtree, namely d. Next comes the root of that subtree, namely b, and then the right subtree in inorder. This is i, followed by the root e, followed by the subtree rooted at j in inorder. This latter listing is m, j, n, o. We continue in this manner, ultimately obtaining: $d, b, i, e, m, j, n, o, a, f, c, g, k, h, p, l$.

13. In postorder, the root comes last, following the left subtree in postorder and the right subtree in postorder. Thus the postorder is the vertices of the left subtree (the one rooted at b) in postorder, then c, then a. Recursively, the postorder in the subtree rooted at b is d, followed by the vertices in the subtree rooted at e in postorder, namely f, g, e, followed by b. Putting this all together, we obtain the answer d, f, g, e, b, c, a.

15. This is just like Exercises 13 and 14. Note that all subtrees of a vertex are completed before listing that vertex. The answer is $k, l, m, e, f, r, s, n, g, b, c, o, h, i, p, q, j, d, a$.

17. **a)** For the first expression, we note that the outermost operation is the second addition. Therefore the root of the tree is this plus sign, and the left and right subtrees are the trees for the expressions being added. The first operand is the sum of x and xy, so the left subtree has a plus sign for its root and the tree for the expressions x and xy as its subtrees. We continue in this manner until we have drawn the entire tree. The second tree is done similarly. Note that the only difference between these two expressions is the placement of parentheses, and yet the expressions represent quite different operations, as can be seen from the fact that the trees are quite different.





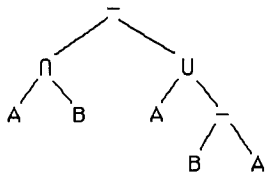
b) We can read off the answer from the picture we have just drawn simply by listing the vertices of the tree in preorder: First list the root, then the left subtree in preorder, then the right subtree in preorder. Therefore the answer is $++x*xy/xy$. Similarly, the second expression in prefix notation is $+x/+*xyxy$.

c) We can read off the answer from the picture we have just drawn simply by listing the vertices of the tree in postorder: First list the left subtree in postorder, then the right subtree in postorder, then the root. Therefore the answer is $xxy*+xy/+$. Similarly, the second expression in postfix notation is $xxy*x+y/+$.

d) The infix expression is just the given expression, fully parenthesized, with an explicit symbol for multiplication. Thus the first is $((x + (x * y)) + (x/y))$, and the second is $(x + (((x * y) + x)/y))$. This corresponds to traversing the tree in inorder, putting in a left parenthesis whenever we go down to a left child and putting in a right parenthesis whenever we come up from a right child.

19. This is similar to Exercise 17, with set operations rather than arithmetic ones.

a) We construct the tree in the same way we did there, noting, for example, that the first minus is the outermost operation.



b) The prefix expression is obtained by traversing the tree in preorder: $- \cap A\, B \cup A - B\, A$.

c) The postfix expression is obtained by traversing the tree in postorder: $A\, B \cap A\, B\, A - \cup -$.

d) This is already in fully parenthesized infix notation except for needing an outer set of parentheses: $((A \cap B) - (A \cup (B - A)))$.

21. Either of the four operators can be the outermost one, so there are four cases to consider. If the first operator is the outermost one, then we need to compute the number of ways to fully parenthesize $B - A \cap B - A$. Here there are 5 possibilities: 1 in which the "$\cap$" symbol is the outermost operator and 2 with each of the "$-$" symbols as the outermost operator. If the second operator in our original expression is the outermost one, then the only choice is in the parenthesization of the second of its operands, and there are 2 possibilities. Thus there are a total 7 ways to parenthesize this expression if either of the first two operators are the outermost one. By symmetry there are another 7 if the outermost operator is one of the last two. Therefore the answer to the problem is 14.

23. We show how to do these exercises by successively replacing the first occurrence of an operator immediately followed by two operands with the result of that operation. (This is an alternative to the method suggested in the text, where the *last* occurrence of an operator, which is necessarily preceded by two operands, is acted upon first.) The final number is the value of the entire prefix expression. In part **(a)**, for example, we first replace / 8 4 by the result of dividing 8 by 4, namely 2, to obtain $- * 2\ 2\ 3$. Then we replace $* 2\ 2$ by the result of multiplying 2 and 2, namely 4, to obtain the third line of our calculation. Next we replace $-\ 4\ 3$ by its answer, 1, which is the final answer.

a)

$$- * 2\ /\ 8\ 4\ 3$$
$$- * 2\ 2\ 3$$
$$-\ 4\ 3$$
$$1$$

b)

$$\uparrow - * 3\ 3 * 4\ 2\ 5$$
$$\uparrow - 9 * 4\ 2\ 5$$
$$\uparrow - 9\ 8\ 5$$
$$\uparrow 1\ 5$$
$$1$$

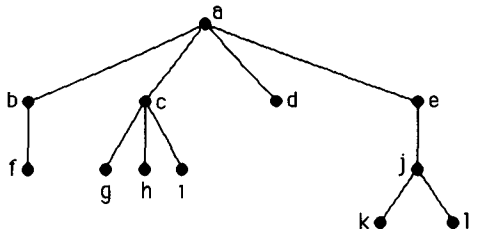
c)

$$+ - \uparrow 3\ 2 \uparrow 2\ 3 \,/\, 6 - 4\ 2$$
$$+ - 9 \uparrow 2\ 3 \,/\, 6 - 4\ 2$$
$$+ - 9\ 8 \,/\, 6 - 4\ 2$$
$$+ 1 \,/\, 6 - 4\ 2$$
$$+ 1 \,/\, 6\ 2$$
$$+ 1\ 3$$
$$4$$

d)

$$* + 3 + 3 \uparrow 3 + 3\ 3\ 3$$
$$* + 3 + 3 \uparrow 3\ 6\ 3$$
$$* + 3 + 3\ 729\ 3$$
$$* + 3\ 732\ 3$$
$$* 735\ 3$$
$$2205$$

25. We slowly use the clues to fill in the details of this tree, shown below. Since the preorder starts with a, we know that a is the root, and we are told that a has four children. Next, since the first child of a comes immediately after a in preorder, we know that this first child is b. We are told that b has one child, and it must be f, which comes next in the preorder. We are told that f has no children, so we are now finished with the subtree rooted at b. Therefore the second child of a must be c (the next vertex in preorder). We continue in this way until we have drawn the entire tree.



27. We prove this by induction on the length of the list. If the list has just one element, then the statement is trivially true. For the inductive step, consider the end of the list. There we find a sequence of vertices, starting with the last leaf and ending with the root of the tree, each vertex being the last child of its successor in the list. We know where this sequence starts, since we are told the number of children of each vertex: it starts at the last leaf in the list. Now remove this leaf, and decrease the child count of its parent by 1. The result is the postorder and child counts of a tree with one fewer vertex. By the inductive hypothesis we can uniquely determine this smaller tree. Then we can uniquely determine where the deleted vertex goes, since it is the last child of its parent (whom we know).

29. In each case the postorder is c, d, b, f, g, h, e, a.

31. We prove this by induction on the recursive definition, in other words, on the length of the formula, i.e., the total number of symbols and operators. The only formula of length 1 arises from the base case of the recursive definition (part (i)), and in that case we have one symbol and no operators, so the statement is true. Assume

CHAPTER 13
Modeling Computation

SECTION 13.1 Languages and Grammars

There is no magical way to come up with the grammars to generate a language described in English. In particular, Exercises 15 and 16 are challenging and very worthwhile. Exercise 21 shows how grammars can be combined. In constructing grammars, we observe the rule that every production must contain at least one nonterminal symbol on the left. This allows us to know when a derivation is completed—namely, when the string we have generated contains no nonterminal symbols.

1. The following sequences of lines show that each is a valid sentence.

a) sentence
noun phrase intransitive verb phrase
article adjective noun intransitive verb phrase
article adjective noun intransitive verb
the **adjective noun intransitive verb**
the happy **noun intransitive verb**
the happy hare **intransitive verb**
the happy hare runs

b) sentence
noun phrase intransitive verb phrase
article adjective noun intransitive verb phrase
article adjective noun intransitive verb adverb
the **adjective noun intransitive verb adverb**
the sleepy **noun intransitive verb adverb**
the sleepy tortoise **intransitive verb adverb**
the sleepy tortoise runs **adverb**
the sleepy tortoise runs quickly

c) sentence
noun phrase transitive verb phrase noun phrase
article noun transitive verb phrase noun phrase
article noun transitive verb noun phrase
article noun transitive verb article noun
the **noun transitive verb article noun**
the tortoise **transitive verb article noun**
the tortoise passes **article noun**
the tortoise passes the **noun**
the tortoise passes the hare

d) sentence
noun phrase transitive verb phrase noun phrase
article adjective noun transitive verb phrase noun phrase
article adjective noun transitive verb noun phrase

article adjective noun transitive verb article adjective noun
the **adjective noun transitive verb article adjective noun**
the sleepy **noun transitive verb article adjective noun**
the sleepy hare **transitive verb article adjective noun**
the sleepy hare passes **article adjective noun**
the sleepy hare passes the **adjective noun**
the sleepy hare passes the happy **noun**
the sleepy hare passes the happy tortoise

3. Since *runs* is only an **intransitive verb**, it can only occur in a sentence of the form **noun phrase intransitive verb phrase**. Such a sentence cannot have anything except an **adverb** after the **intransitive verb**, and *the sleepy tortoise* cannot be an **adverb**.

5. **a)** It suffices to give a derivation of this string. We write the derivation in the obvious way. $S \Rightarrow 1A \Rightarrow 10B \Rightarrow 101A \Rightarrow 1010B \Rightarrow 10101$.

b) This follows from our solution to part **(c)**, because 10110 has two 1's in a row and is not of the form discussed there.

c) Notice that the only production with A on the left is $A \rightarrow 0B$. Furthermore, the only productions with B on the left are $B \rightarrow 1A$ and $B \rightarrow 1$. Combining these, we see that we can eliminate B and replace these three rules by $A \rightarrow 01A$ and $A \rightarrow 01$. This tells us that every string in the language generated by G must end with some number of repetitions of 01 (at least one). Furthermore, because of the rules $S \rightarrow 0A$ and $S \rightarrow 1A$, the string must start with either a 0 or a 1 preceding the repetitions of 01. Therefore the strings in this language consist of a 0 or a 1 followed by one or more repetitions of 01. We can write this as $\{0(01)^n \mid n \geq 0\} \cup \{1(01)^n \mid n \geq 0\}$

7. We write the derivation in the obvious way. $S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 000S111 \Rightarrow 000111$. We used the rule $S \rightarrow 0S1$ in the first three steps and $S \rightarrow \lambda$ in the last step.

9. **a)** Using G_1, we can add 0's on the left or 1's on the right of S. Thus we have $S \Rightarrow 0S \Rightarrow 00S \Rightarrow 00S1 \Rightarrow 00S11 \Rightarrow 00S111 \Rightarrow 00S1111 \Rightarrow 001111$.

b) In this grammar we must add all the 0's first to S, then change to an A and add the 1's, again on the left. Thus we have $S \Rightarrow 0S \Rightarrow 00S \Rightarrow 001A \Rightarrow 0011A \Rightarrow 00111A \Rightarrow 001111$.

11. First we apply the first rule twice and the rule $S \rightarrow \lambda$ to get $00ABAB$. We can then apply the rule $BA \rightarrow AB$, to get $00AABB$. Now we can apply the rules $0A \rightarrow 01$ and $1A \rightarrow 11$ to get $0011BB$; and then the rules $1B \rightarrow 12$ and $2B \rightarrow 22$ to end up with 001122, as desired.

13. In each case we will list only the productions, because V and T will be obvious from the context, and S speaks for itself.

a) For this finite set of strings, we can simply have $S \rightarrow 0$, $S \rightarrow 1$, and $S \rightarrow 11$.

b) We assume that "only 1's" includes the case of no 1's. Thus we can take simply $S \rightarrow 1S$ and $S \rightarrow \lambda$.

c) The middle can be anything we like, and we will let A represent the middle. Then our productions are $S \rightarrow 0A1$, $A \rightarrow 1A$, $A \rightarrow 0A$, and $A \rightarrow \lambda$.

d) We will let A represent the pairs of 1's. Then our productions are $S \rightarrow 0A$, $A \rightarrow 11A$, and $A \rightarrow \lambda$.

15. **a)** We need to add the 0's two at a time. Thus we can take the rules $S \rightarrow S00$ and $S \rightarrow \lambda$.

b) We can use the same first rule as in part **(a)**, namely $S \rightarrow S00$, to increase the number of 0's. Since the string must begin 10, we simply adjoin to this the rule $S \rightarrow 10$.

c) We need to add 0's and 1's two at a time. Furthermore, we need to allow for 0's and 1's to change their order. Since we cannot have a rule $01 \rightarrow 10$ (there being no nonterminal symbol on the left), we make up nonterminal analogs of 0 and 1, calling them A and B, respectively. Thus our rules are as follows: $S \rightarrow AAS$, $S \rightarrow BBS$, $AB \rightarrow BA$, $BA \rightarrow AB$, $S \rightarrow \lambda$, $A \rightarrow 0$, and $B \rightarrow 1$. (There are also totally different ways to approach this problem, which are just as effective.)

d) This one is fairly simple: $S \rightarrow 0000000000A$, $A \rightarrow 0A$, $A \rightarrow \lambda$. This assures at least 10 0's and allows for any number of additional 0's.

e) We need to invoke the trick used in part **(c)** to allow 0's and 1's to change their order. Furthermore, since we need at least one extra 0, we use $S \rightarrow A$ as our vanishing condition, rather than $S \rightarrow \lambda$. Our solution, then, is $S \rightarrow AS$, $S \rightarrow ABS$, $S \rightarrow A$, $AB \rightarrow BA$, $BA \rightarrow AB$, $A \rightarrow 0$, and $B \rightarrow 1$.

f) This is identical to part **(e)**, except that the vanishing condition is $S \rightarrow \lambda$, rather than $S \rightarrow A$, and there is no rule $S \rightarrow AS$.

g) We just put together two copies of a solution to part **(e)**, one in which there are more 0's than 1's, and one in which there are more 1's than 0's. The rules are as follows: $S \rightarrow ABS$, $S \rightarrow T$, $S \rightarrow U$, $T \rightarrow AT$, $T \rightarrow A$, $U \rightarrow BU$, $U \rightarrow B$, $AB \rightarrow BA$, $BA \rightarrow AB$, $A \rightarrow 0$, and $B \rightarrow 1$.

17. In each case we will list only the productions, because V and T will be obvious from the context, and S speaks for itself.

a) It suffices to have $S \rightarrow 0S$ and $S \rightarrow \lambda$.

b) We let A represent the string of 1's. Thus we take $S \rightarrow A0$, $A \rightarrow 1A$, and $A \rightarrow \lambda$. Notice that $A \rightarrow A1$ works just as well $A \rightarrow 1A$ here, so either one is fine.

c) It suffices to have $S \rightarrow 000S$ and $S \rightarrow \lambda$.

19. a) This is a type 2 grammar, because the left-hand side of each production has a single nonterminal symbol. It is not a type 3 grammar, because the right-hand side of the productions are not of the required type.

b) This meets the definition of a type 3 grammar.

c) This is only a type 0 grammar; it does not fit the definition of type 1 because the right side of the second production does not maintain the context set by the left side.

d) This is a type 2 grammar, because the left-hand side of each production has a single nonterminal symbol. It is not a type 3 grammar, because the right-hand side of the productions are not of the required type.

e) This meets the definition of a type 2 grammar. It is not of type 3, because of the production $A \rightarrow B$.

f) This is only a type 0 grammar; it does not fit the definition of type 1 because the right side of the second production does not maintain the context set by the left side.

g) This meets the definition of a type 3 grammar. Note, however, that it does not meet the definition of a type 1 grammar because of $S \rightarrow \lambda$.

h) This is only a type 0 grammar; it does not fit the definition of type 1 because the right side of the third production does not maintain the context set by the left side.

i) This is a type 2 grammar because each left-hand side is a single nonterminal. It is not type 3 because of the production $B \rightarrow \lambda$.

j) This is a type 2 grammar because each left-hand side is a single nonterminal. It is not type 3; each of the productions violates the conditions imposed for a type 3 grammar.

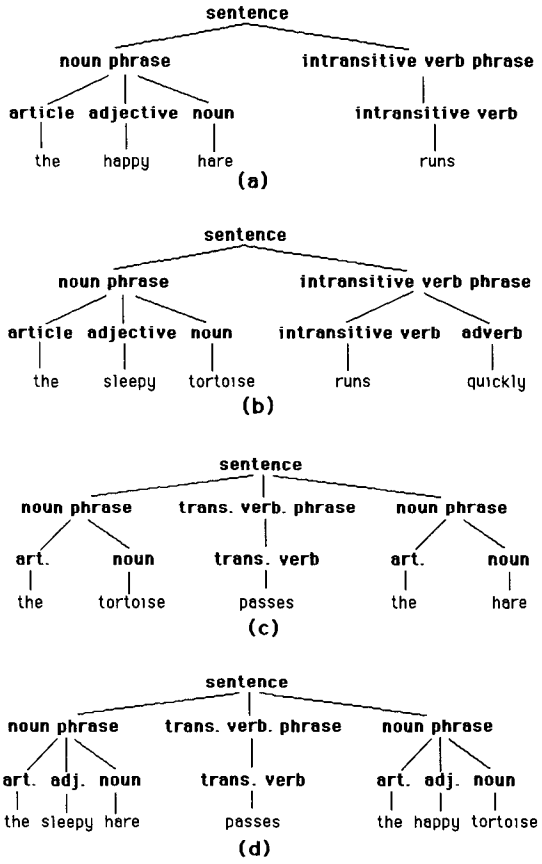
21. Let us assume that the nonterminal symbols of G_1 and G_2 are disjoint. (If they are not, we can give those in G_2, say, new names so that they will be; obviously this does not change the language that G_2 generates.) Call the start symbols S_1 and S_2. In each case we will define G by taking all the symbols and rules for G_1 and G_2, a new symbol S, which will be the start symbol for G, and the rules listed below.

a) Since we want strings that either G_1 or G_2 generate, we add the rules $S \rightarrow S_1$ and $S \rightarrow S_2$.

b) Since we want strings that consist of a string that G_1 generates followed by a string that G_2 generates, we add the rule $S \rightarrow S_1 S_2$.

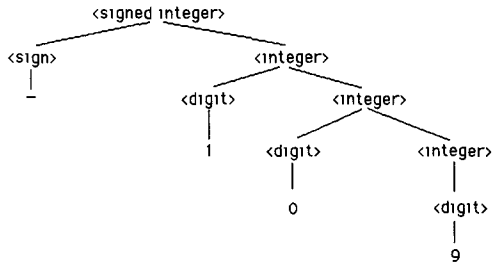
c) This time we add the rules $S \rightarrow S_1 S$ and $S \rightarrow \lambda$. This clearly gives us all strings that consist of the concatenation of any number of strings that G_1 generates.

23. We simply translate the derivations we gave in the solution to Exercise 1 to tree form, obtaining the following pictures.



25. We can assume that the derivation starts $S \Rightarrow AB \Rightarrow CaB \Rightarrow cbaB$, or $S \Rightarrow AB \Rightarrow CaB \Rightarrow baB$. This shows that neither the string in part **(b)** nor the string in part **(d)** is in the language, since they do not begin cba or ba. In order to derive the string in part **(a)**, we need to turn B into ba, and this is easy, using the rule $B \rightarrow Ba$ and then the rule $B \rightarrow b$. Finally, for part **(c)**, we again simply apply these two rules to change B into ba.

27. This is straightforward. The $-$ is the sign and the 109 is an integer, so the tree starts as shown. Then we decompose the integer 109 into the digit 1 and the integer 09, then in turn to the digit 0 and the integer (digit) 9.



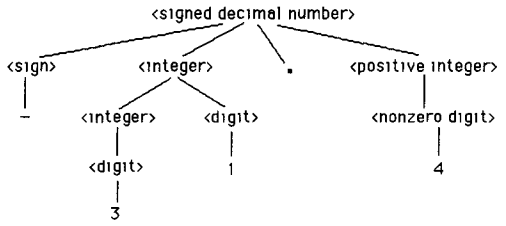
29. a) Note that a string such as "34." is not allowed by this definition, but a string such as -02.780 is. This is pretty straightforward using the following rules. As can be seen, we are using $\langle integer \rangle$ to stand for a nonnegative integer.

$S \rightarrow \langle sign \rangle \langle integer \rangle$
$S \rightarrow \langle sign \rangle \langle integer \rangle . \langle positive\ integer \rangle$
$\langle sign \rangle \rightarrow +$
$\langle sign \rangle \rightarrow -$
$\langle integer \rangle \rightarrow \langle integer \rangle \langle digit \rangle$
$\langle integer \rangle \rightarrow \langle digit \rangle$
$\langle positive\ integer \rangle \rightarrow \langle integer \rangle \langle nonzero\ digit \rangle \langle integer \rangle$
$\langle positive\ integer \rangle \rightarrow \langle integer \rangle \langle nonzero\ digit \rangle$
$\langle positive\ integer \rangle \rightarrow \langle nonzero\ digit \rangle \langle integer \rangle$
$\langle positive\ integer \rangle \rightarrow \langle nonzero\ digit \rangle$
$\langle digit \rangle \rightarrow \langle nonzero\ digit \rangle$
$\langle digit \rangle \rightarrow 0$
$\langle nonzero\ digit \rangle \rightarrow 1$
$\langle nonzero\ digit \rangle \rightarrow 2$
$\langle nonzero\ digit \rangle \rightarrow 3$
$\langle nonzero\ digit \rangle \rightarrow 4$
$\langle nonzero\ digit \rangle \rightarrow 5$
$\langle nonzero\ digit \rangle \rightarrow 6$
$\langle nonzero\ digit \rangle \rightarrow 7$
$\langle nonzero\ digit \rangle \rightarrow 8$
$\langle nonzero\ digit \rangle \rightarrow 9$

b) We combine rows of the previous answer with the same left-hand side, and we change the notation to produce the answer to this part.

$\langle signed\ decimal\ number \rangle ::= \langle sign \rangle \langle integer \rangle \mid \langle sign \rangle \langle integer \rangle . \langle positive\ integer \rangle$
$\langle sign \rangle ::= + \mid -$
$\langle integer \rangle ::= \langle integer \rangle \langle digit \rangle \mid \langle digit \rangle$
$\langle positive\ integer \rangle ::= \langle integer \rangle \langle nonzero\ digit \rangle \langle integer \rangle \mid \langle integer \rangle \langle nonzero\ digit \rangle$
$\quad \mid \langle nonzero\ digit \rangle \langle integer \rangle \mid \langle nonzero\ digit \rangle$
$\langle digit \rangle ::= \langle nonzero\ digit \rangle \mid 0$
$\langle nonzero\ digit \rangle ::= 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

c) We easily produce the following tree.



31. a) We can think of appending letters to the end at each stage:

$\langle identifier \rangle ::= \langle lcletter \rangle \mid \langle identifier \rangle \langle lcletter \rangle$
$\langle lcletter \rangle ::= a \mid b \mid c \mid \cdots \mid z$

b) We need to be more explicit here than in part **(a)** about how many letters are used:

$\langle identifier \rangle ::= \langle lcletter \rangle \langle lcletter \rangle \langle lcletter \rangle \mid \langle lcletter \rangle \langle lcletter \rangle \langle lcletter \rangle \langle lcletter \rangle \mid$
$\quad \langle lcletter \rangle \langle lcletter \rangle \langle lcletter \rangle \langle lcletter \rangle \langle lcletter \rangle \mid$

$\langle lcletter\rangle\langle lcletter\rangle\langle lcletter\rangle\langle lcletter\rangle\langle lcletter\rangle\langle lcletter\rangle$

$\langle lcletter\rangle ::= a \mid b \mid c \mid \cdots \mid z$

c) This is similar to the part **(b)**, allowing for two types of letters:

$\langle identifier\rangle ::= \langle ucletter\rangle \mid \langle ucletter\rangle\langle letter\rangle \mid \langle ucletter\rangle\langle letter\rangle\langle letter\rangle \mid$
$\langle ucletter\rangle\langle letter\rangle\langle letter\rangle\langle letter\rangle \mid \langle ucletter\rangle\langle letter\rangle\langle letter\rangle\langle letter\rangle\langle letter\rangle \mid$
$\langle ucletter\rangle\langle letter\rangle\langle letter\rangle\langle letter\rangle\langle letter\rangle\langle letter\rangle$

$\langle letter\rangle ::= \langle lcletter\rangle \mid \langle ucletter\rangle$

$\langle lcletter\rangle ::= a \mid b \mid c \mid \cdots \mid z$

$\langle ucletter\rangle ::= A \mid B \mid C \mid \cdots \mid Z$

d) This is again similar to previous parts. We need to invent a name for "digit or underscore."

$\langle identifier\rangle ::= \langle lcletter\rangle\langle digitorus\rangle\langle alphanumeric\rangle\langle alphanumeric\rangle\langle alphanumeric\rangle \mid$
$\langle lcletter\rangle\langle digitorus\rangle\langle alphanumeric\rangle\langle alphanumeric\rangle\langle alphanumeric\rangle\langle alphanumeric\rangle$

$\langle digitorus\rangle ::= \langle digit\rangle \mid \_$

$\langle alphanumeric\rangle ::= \langle letter\rangle \mid \langle digit\rangle$

$\langle letter\rangle ::= \langle lcletter\rangle \mid \langle ucletter\rangle$

$\langle lcletter\rangle ::= a \mid b \mid c \mid \cdots \mid z$

$\langle ucletter\rangle ::= A \mid B \mid C \mid \cdots \mid Z$

$\langle digit\rangle ::= 0 \mid 1 \mid 2 \mid \cdots \mid 9$

33. We create a name for "letter or underscore" and then define an identifier to consist of one of those, followed by any number of other allowed symbols. Note that an underscore by itself is a valid identifier, and there is no prohibition on consecutive underscores.

$\langle identifier\rangle ::= \langle letterorus\rangle \mid \langle identifier\rangle\langle symbol\rangle$

$\langle symbol\rangle ::= \langle letterorus\rangle \mid \langle digit\rangle$

$\langle letterorus\rangle ::= \langle letter\rangle \mid \_$

$\langle letter\rangle ::= \langle lcletter\rangle \mid \langle ucletter\rangle$

$\langle lcletter\rangle ::= a \mid b \mid c \mid \cdots \mid z$

$\langle ucletter\rangle ::= A \mid B \mid C \mid \cdots \mid Z$

$\langle digit\rangle ::= 0 \mid 1 \mid 2 \mid \cdots \mid 9$

35. We assume that leading 0's are not allowed in the whole number part, since the problem explicitly mentioned them only for the decimal part. Our rules have to allow the optional sign using the question mark, the integer part consisting of one or more digits, not beginning with a 0 unless 0 is the entire whole number part, and then either the decimal part or not. Note that the decimal part has a decimal point followed by zero or more digits.

$numeral ::= sign?\ nonzerodigit\ digit*\ decimal? \mid sign?\ 0\ decimal?$

$decimal ::= .digit*$

$digit ::= 0 \mid nonzerodigit$

$sign ::= + \mid -$

$nonzerodigit ::= 1 \mid 2 \mid \cdots \mid 9$

37. We can simplify the answer given in Exercise 33 using the asterisk for repeating optional elements.

$identifier ::= letterorus\ symbol*$

$symbol ::= letterorus \mid digit$

$letterorus ::= letter \mid \_$

$letter ::= lcletter \mid ucletter$

$lcletter ::= a \mid b \mid c \mid \cdots \mid z$

$ucletter ::= A \mid B \mid C \mid \cdots \mid Z$

$digit ::= 0 \mid 1 \mid 2 \mid \cdots \mid 9$

39. **a)** This string is generated by the grammar. The substring $bc*$ is a term, since it consists of the factor b followed by the factor c followed by the mulOperator $*$. Thus the entire expression consists of two terms followed by an addOperator. We can show the steps in the following sequence:

$\langle expression \rangle$
$\langle term \rangle \langle term \rangle \langle addOperator \rangle$
$\langle factor \rangle \langle factor \rangle \langle factor \rangle \langle mulOperator \rangle \langle addOperator \rangle$
$\langle identifier \rangle \langle identifier \rangle \langle identifier \rangle \langle mulOperator \rangle \langle addOperator \rangle$
$a\,b\,c\,*\,+$

b) This string is not generated by the grammar. The second plus sign needs two terms preceding it, and $xy+$ can only be deconstructed to be one term.

c) This string is generated by the grammar. The substring $xy-$ is a factor, since it is an expression, namely the term x followed by the term y followed by the addOperator $-$. Thus the entire expression consists of two factors followed by a mulOperator. We can show the steps in the following sequence:

$\langle expression \rangle$
$\langle term \rangle$
$\langle factor \rangle \langle factor \rangle \langle mulOperator \rangle$
$\langle expression \rangle \langle factor \rangle \langle mulOperator \rangle$
$\langle term \rangle \langle term \rangle \langle addOperator \rangle \langle factor \rangle \langle mulOperator \rangle$
$\langle factor \rangle \langle factor \rangle \langle addOperator \rangle \langle factor \rangle \langle mulOperator \rangle$
$\langle identifier \rangle \langle identifier \rangle \langle addOperator \rangle \langle identifier \rangle \langle mulOperator \rangle$
$x\,y\,-\,z\,*$

d) This is similar to part **(c)**. The entire expression consists of two factors followed by a mulOperator; the first of these factors is just w, and the second is the term $x\,y\,z\,-\,*$. That term, in turn, deconstructs as in previous parts. We can show the steps in the following sequence:

$\langle expression \rangle$
$\langle term \rangle$
$\langle factor \rangle \langle factor \rangle \langle mulOperator \rangle$
$\langle factor \rangle \langle expression \rangle \langle mulOperator \rangle$
$\langle factor \rangle \langle term \rangle \langle mulOperator \rangle$
$\langle factor \rangle \langle factor \rangle \langle factor \rangle \langle mulOperator \rangle \langle mulOperator \rangle$
$\langle factor \rangle \langle factor \rangle \langle expression \rangle \langle mulOperator \rangle \langle mulOperator \rangle$
$\langle factor \rangle \langle factor \rangle \langle term \rangle \langle term \rangle \langle addOperator \rangle \langle mulOperator \rangle \langle mulOperator \rangle$
$\langle factor \rangle \langle factor \rangle \langle factor \rangle \langle factor \rangle \langle addOperator \rangle \langle mulOperator \rangle \langle mulOperator \rangle$
$\langle identifier \rangle \langle identifier \rangle \langle identifier \rangle \langle identifier \rangle \langle addOperator \rangle \langle mulOperator \rangle \langle mulOperator \rangle$
$w\,x\,y\,z\,-\,*\,/$

e) This string is generated as follows (similar to previous parts of this exercise):

$\langle expression \rangle$
$\langle term \rangle$
$\langle factor \rangle \langle factor \rangle \langle mulOperator \rangle$
$\langle factor \rangle \langle expression \rangle \langle mulOperator \rangle$
$\langle factor \rangle \langle term \rangle \langle term \rangle \langle addOperator \rangle \langle mulOperator \rangle$
$\langle factor \rangle \langle factor \rangle \langle factor \rangle \langle addOperator \rangle \langle mulOperator \rangle$
$\langle identifier \rangle \langle identifier \rangle \langle identifier \rangle \langle addOperator \rangle \langle mulOperator \rangle$
$a\,d\,e\,-\,*$

41. The answers will depend on the grammar given as the solution to Exercise 40. We assume here that the answer to that exercise is very similar to the preamble to Exercise 39. The only difference is that the operators are

placed between their operands, rather than behind them, and parentheses are required in expressions used as factors.

a) This string is not generated by the grammar, because the addition operator can only be applied to two terms, and terms that are themselves expressions must be surrounded by parentheses.

b) This string is generated by the grammar. The substrings a/b and c/d are terms, so they can be combined to form the expression. We show the steps in the following sequence:

$\langle expression \rangle$
$\langle term \rangle \langle addOperator \rangle \langle term \rangle$
$\langle factor \rangle \langle mulOperator \rangle \langle factor \rangle \langle addOperator \rangle \langle factor \rangle \langle mulOperator \rangle \langle factor \rangle$
$\langle identifier \rangle \langle mulOperator \rangle \langle identifier \rangle \langle addOperator \rangle \langle identifier \rangle \langle mulOperator \rangle \langle identifier \rangle$
$a/b + c/d$

c) This string is generated by the grammar. The substring $(n + p)$ is a factor, since it is an expression surrounded by parentheses. We show the steps in the following sequence:

$\langle expression \rangle$
$\langle term \rangle$
$\langle factor \rangle \langle mulOperator \rangle \langle factor \rangle$
$\langle factor \rangle \langle mulOperator \rangle (\langle expression \rangle)$
$\langle factor \rangle \langle mulOperator \rangle (\langle term \rangle \langle addOperator \rangle \langle term \rangle)$
$\langle factor \rangle \langle mulOperator \rangle (\langle factor \rangle \langle addOperator \rangle \langle factor \rangle)$
$\langle identifier \rangle \langle mulOperator \rangle (\langle identifier \rangle \langle addOperator \rangle \langle identifier \rangle)$
$m * (n + p)$

d) There are several reasons that this string is not generated, among them the fact that it is impossible for an expression to start with an operator in this grammar.

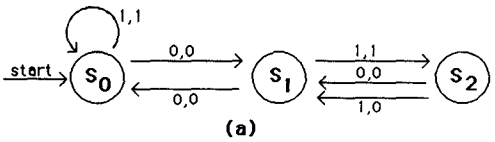
e) This is very similar to part **(c)**:

$\langle expression \rangle$
$\langle term \rangle$
$\langle factor \rangle \langle mulOperator \rangle \langle factor \rangle$
$(\langle expression \rangle) \langle mulOperator \rangle (\langle expression \rangle)$
$(\langle term \rangle \langle addOperator \rangle \langle term \rangle) \langle mulOperator \rangle (\langle term \rangle \langle addOperator \rangle \langle term \rangle)$
$(\langle factor \rangle \langle addOperator \rangle \langle factor \rangle) \langle mulOperator \rangle (\langle factor \rangle \langle addOperator \rangle \langle factor \rangle)$
$(\langle identifier \rangle \langle addOperator \rangle \langle identifier \rangle) \langle mulOperator \rangle (\langle identifier \rangle \langle addOperator \rangle \langle identifier \rangle)$
$(m + n) * (p - q)$

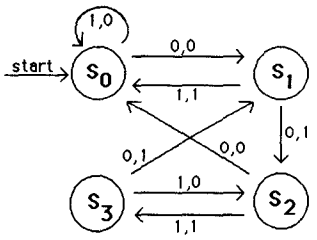
SECTION 13.2 Finite-State Machines with Output

Finding finite-state machines to do specific tasks is in essence computer programming. There is no set method for doing this. You have to think about the problem for awhile, ask yourself what it might be useful for the states to represent, and then very carefully proceed to construct the machine. Expect to have several false starts. "Bugs" in your machines are also very common. There are of course many machines that will accomplish the same task. The reader should look at Exercises 20–25 to see that it is also possible to build finite-state machines with the output associated with the states, rather than the transitions.

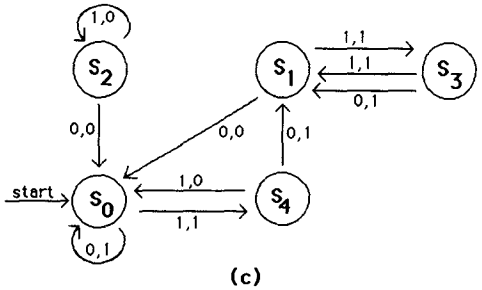
1. We draw the state diagrams by making a node for each state and a labeled arrow for each transition. In part **(a)**, for example, since under input 1 from state s_2 we are told that we move to state s_1 and output a 0, we draw an arrow from s_2 to s_1 and label it $1,0$. It is assumed that s_0 is always the start state.



(a)



(b)



(c)

3. **a)** The machine starts in state s_0. Since the first input symbol is 0, the machine moves to state s_1 and gives 0 as output. The next input symbol is 1, so the machine moves to state s_2 and gives 1 as output. The next input is 1, so the machine moves to state s_1 and gives 0 as output. The fourth input is 1, so the machine moves to state s_2 and gives 1 as output. The fifth input is 0, so the machine moves to state s_1 and gives 0 as output. Thus the output is 01010.

b) The machine starts in state s_0. Since the first input symbol is 0, the machine moves to state s_1 and gives 0 as output. The next input symbol is 1, so the machine moves to state s_0 and gives 1 as output. The next input is 1, so the machine stays in state s_0 and gives 0 as output. The fourth input is 1, so the machine again stays in state s_0 and gives 0 as output. The fifth input is 0, so the machine moves to state s_1 and gives 0 as output. Thus the output is 01000.

c) The machine starts in state s_0. Since the first input symbol is 0, the machine stays in state s_0 and gives 1 as output. The next input symbol is 1, so the machine moves to state s_4 and gives 1 as output. The next input is 1, so the machine moves to state s_0 and gives 0 as output. The fourth input is 1, so the machine moves to state s_4 and gives 1 as output. The fifth input is 0, so the machine moves to state s_1 and gives 1 as output. Thus the output is 11011.

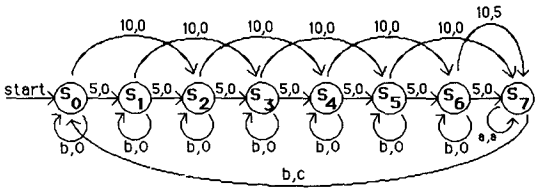
5. **a)** The machine starts in state s_0. Since the first input symbol is 0, the machine moves to state s_1 and gives 1 as output. (This is what the arrow from s_0 to s_1 with label $0,1$ means.) The next input symbol is 1. Because of the edge from s_1 to s_0, the machine moves to state s_0 and gives 1 as output. The next input is 1. Because of the loop at s_0, the machine stays in state s_0 and gives output 0. The same thing happens on the fourth input symbol. Therefore the output is 1100 (and the machine ends up in state s_0).

b) This is similar to part **(a)**. The first two symbols of input cause the machine to output two 0's and remain in state s_0. The third symbol causes an output of 1 as the machine moves into state s_1. The fourth input

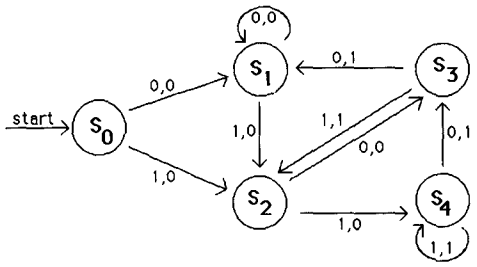
takes us back to state s_0 with output 1. The next four symbols of input cause the machine to give output 0110 as it goes to states s_0, s_1, s_0, and s_0, respectively. Therefore the output is 00110110.

c) This is similar to the other parts. The machine alternates between states s_0 and s_1, outputting 1 for each input. Thus the output is 11111111111.

7. We model this machine as follows. There are four possible inputs, which we denote by 5, 10, 25, and b, standing for a nickel, a dime, a quarter, and a button labeled by a kind of soda pop, respectively. (Actually the model is a bit more complicated, since there are three kinds of pop, but we will ignore that; to incorporate the kind of pop into the model, we would simply have three inputs in place of just b.) The output can either be an amount of money in cents—0, 5, 10, 15, 20, or 25—or can be a can of soda pop, which we denote c. There will be eight states. Intuitively, state s_i will represent the state in which the machine is indebted to the customer by $5i$ cents. Thus s_0, the start state, will represent that the machine owes the customer nothing; state s_1 will represent that the machine has accepted 5 cents from the customer, and so on. State s_7 will mean that the machine owes the customer 35 cents, which will be paid with a can of soda pop, at which time the machine will return to state s_0, owing nothing. The following picture is the state diagram of this machine, simplified even further in that we have eliminated quarters entirely for sake of readability. For example, the transition from state s_6 (30 cents credit) on input of a dime is to state s_7 (35 cents credit) with the return of 5 cents in change. We have also used a to stand for any monetary input: if you deposit any amount when the machine already has your 35 cents, then you get that same amount back. Thus the transition a, a really stands for three transitions: $5, 5$ and $10, 10$ and $25, 25$.

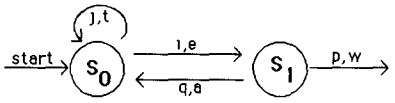


9. We draw the diagram for this machine. Intuitively, we need four states, corresponding to the four possibilities for what the last two bits have been. In our picture, state s_1 corresponds to the last two bits having been 00; state s_2 corresponds to the last two bits having been 01; state s_3 corresponds to the last two bits having been 10; state s_4 corresponds to the last two bits having been 11. We also need a state s_0 to get started, to account for the delay. Let us see why some of the transitions are what they are. If you are in state s_3, then the last two bits have been 10. If you now receive an input 0, then the last two bits will be 00, so we need to move to state s_1. Furthermore, since the bit received two pulses ago was a 1 (we know this from the fact that we are in state s_3), we need to output a 1. Also, since we are told to output 00 at the beginning, it is right to have transitions from s_0 as shown.

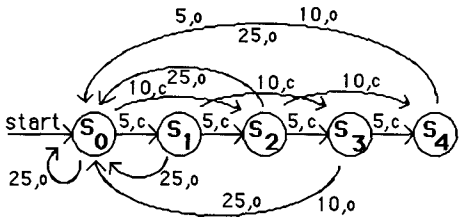


If we look at this machine, we observe that states s_0 and s_1 are equivalent, i.e., they cause exactly the same transitions and outputs. Therefore a simpler answer would be a machine like this one, but without state s_0, where state s_1 is the start state.

11. This machine is really only part of a machine; we are not told what happens after a successful log-on. Also, the machine is really much more complicated than we are indicating here, because we really need a separate state for each user. We assume that there is only one user. We also assume that an invalid user ID is rejected immediately, without a request for a password. (The alternate assumption is also reasonable, that the machine requests a password whether or not the ID is valid. In that case we obtain a different machine, of course.) We need only two states. The initial state waits for the valid user ID. We let i be the valid user ID, and we let j be any other input. If the input is valid, then we enter state s_1, outputting the message e: "enter your password." If the input is not valid, then we remain in state s_0, outputting the message t: "invalid ID; try again." From state s_1 there are only two relevant inputs: the valid password p and any other input q. If the input is valid, then we output the message w: "welcome" and proceed. If the input is invalid, then we output the message a: "invalid password; enter user ID again" and return to state s_0 to await another attempt at logging-on.



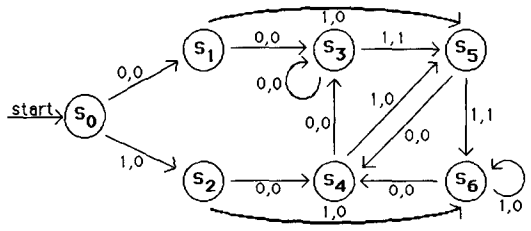
13. This exercise is similar to Exercise 7. We let state s_i for $i = 0, 1, 2, 3, 4$ represent the fact that $5i$ cents has been deposited. When at least 25 cents has been deposited, we return to state s_0 and open the gate. Nickels (input 5), dimes (input 10) and quarters (input 25) are available. We let o and c be the outputs: the gate is opened (for a limited time, of course), or remains closed. After the gate is opened, we return to state s_0. (We assume that the gate closes after the car has passed.)



15. The picture for this machine would be too complex to draw. Instead, we will describe the machine verbally, and even then we won't give every last gory detail. We assume that possible inputs are the ten digits. We will let s_0 be the start state and let s_1 be the state representing a successful call (so we will not list any outputs from s_1). From s_0, inputs of 2, 3, 4, 5, 6, 7, or 8 send the machine back to s_0 with output of an error message for the user. From s_0 an input of 0 sends the machine to state s_1, with the output being that the 0 is sent to the network. From s_0 an input of 9 sends the machine to state s_2 with no output; from there an input of 1 sends the machine to state s_3 with no output; from there an input of 1 sends the machine to state s_1 with the output being that the 911 is sent to the network. All other inputs while in states s_2 or s_3 send the machine back to s_0 with output of an error message for the user. From s_0 an input of 1 sends the machine to state s_4 with no output; from s_4 an input of 2 sends the machine to state s_5 with no output; and this path continues in a similar manner to the 911 path, looking next for 1, then 2, then any seven digits, at which point the machine goes to state s_1 with the output being that the ten-digit input is sent to the network. Any "incorrect" input while in states s_5 or s_6 (that is, anything except a 1 while in s_5 or a 2 while in s_6) sends the machine back to s_0 with output of an error message for the user. Similarly, from s_4 an input of 8 followed by appropriate successors drives us eventually to s_1, but inappropriate outputs drive us back to s_0 with an error message. Also, inputs while in state s_4 other than 2 or 8 send the machine back s_0 with output of an error message for the user.

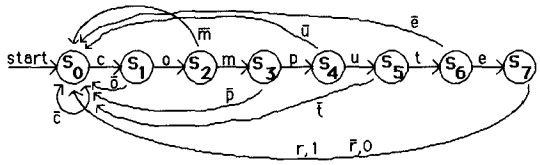
17. We interpret this problem as asking that a 1 be output if the conditions are met, and a 0 be output otherwise. For this machine, we need to keep track of what the last two inputs have been, and we need four states to

"store" this information. Let the states s_3, s_4, s_5, and s_6 be the states corresponding to the last two inputs having been 00, 10, 01, and 11, respectively. We also need some states to get started—to get us into one of these four states. There are only two cases in which the output is 1: if we are in states s_3 or s_5 (so that the last two inputs have been 00 or 01) and we receive a 1 as input. The transitions in our machine are the obvious ones. For example, if we are in state s_5, having just read 01, and receive a 0 as input, then the last two symbols read are now 10, so we move to state s_4.



As in Exercise 9, we can actually get by with a smaller machine. Note that here states s_1 and s_4 are equivalent, as are states s_2 and s_6. Thus we can merge each of these pairs into one state, producing a machine with only five states. At that point, furthermore, state s_0 is equivalent to the merged s_2 and s_6, so we can omit state s_0 and make this other state the start state. The reader is urged to draw the diagram for this simpler machine.

19. We need some notation to make our picture readable. The alphabet has 26 symbols in it. If α is a letter, then by $\overline{\alpha}$ we mean any letter other than α. Thus an arrow labeled $\overline{\alpha}$ really stands for 25 arrows. The output is to be 1 when we have just finished reading the word *computer*. Thus we need eight states, to stand for the various stages of having read part of that word. The picture below gives the details, except that we have omitted all the outputs except on inputs r and $\overline{r}$; all the omitted ones are intended to be 0. The reader might contemplate why this problem would have been harder if the word in question were something like *baboon*.



21. We construct the state table by having one row for each state. The arrows tell us what the values of the transition function are. For example, since there is an arrow from s_0 to s_1 labeled 0, the transition from s_0 on input 0 is to s_1. Similarly, the transition from s_0 on input 1 is to s_2. The output function values are shown next to each state. Thus the output for state s_0 is 1, the output for state s_1 is 1, and the output for state s_2 is 0. The table is therefore as shown here.

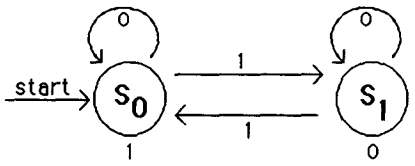
	Input		
State	0	1	Output
s_0	s_1	s_2	1
s_1	s_1	s_0	1
s_2	s_1	s_2	0

23. **a)** The input drives the machine successively to states s_1, s_0, s_1, and s_0. The output is the output of the start state, followed by the outputs of these four states, namely 11111.

b) The input drives the machine to state s_2, where it remains because of the loop. The output is the output of the start state, followed by the output at state s_2 six times, namely 1000000.

c) The states visited after the start state are, in order, s_2, s_2, s_2, s_1, s_0, s_2, s_2, s_1, s_0, s_2, and s_2. Therefore the output is 100011001100.

25. We can use a machine with just two states, one to indicate that there is an even number of 1's in the input string, the other to indicate that there is an odd number of 1's in the string. Since the empty string has an even number of 1's, we make s_0 (the start state) the state for an even number of 1's. The output for this state will be 1, as directed. The output from state s_1 will be 0 to indicate an odd number of 1's. The input 1 will drive the machine from one state to the other, while the input 0 will keep the machine in its current state. The diagram below gives the desired machine.



SECTION 13.3 Finite-State Machines with No Output

As in the previous section, many of these exercises are really exercises in programming. There is no magical way to become a good programmer, but experience helps. The converse problem is also hard—finding a good verbal description of the set recognized by a given finite-state automaton.

1. **a)** This is the set of all strings ab, where $a \in A$ and $b \in B$. Thus it contains precisely 000, 001, 1100, and 1101.

b) This is the set of all strings ba, where $a \in A$ and $b \in B$. Thus it contains precisely 000, 0011, 010, and 0111.

c) This is the set of all strings a_1a_2, where $a_1 \in A$ and $a_2 \in A$. Thus it contains precisely 00, 011, 110, and 1111.

d) This is the set of all strings $b_1b_2b_3$, where each $b_i \in B$. Thus it contains precisely 000000, 000001, 000100, 000101, 010000, 010001, 010100 and 010101.

3. Two possibilities are of course to let A be this entire set and let $B = \{\lambda\}$, and to let B be this entire set and let $A = \{\lambda\}$. Let us find more. With a little experimentation we see that $A = \{\lambda, 10\}$ and $B = \{10, 111, 1000\}$ also works, and it can be argued that there are no other solutions in which λ appears in either set. Finally, there is the solution $A = \{1, 101\}$ and $B = \{0, 11, 000\}$. It can be argued that there are no more. (Here is how the first of these arguments goes. If $\lambda \in A$, then necessarily $\lambda \notin B$. Hence the shortest string in B has length at least 2, from which it follows that $10 \in B$. Now since the only other string in AB that ends with 10 is 1010, the only possible other string in A is 10. This leads to the third solution mentioned above. On the other hand, if $\lambda \in B$, then $\lambda \notin A$, so it must be that the shortest string in A is 10. This forces 111 to be in A, and now there can be no other strings in B. The second argument is similar.)

5. **a)** One way to write this answer is $\{(10)^n \mid n = 0, 1, 2, \dots\}$. It is the concatenation of zero or more copies of the string 10.

b) This is like part **(a)**. This set consists of all copies of zero or more concatenations of the string 111. In other words, it is the set of all strings of 1's of length a multiple of 3. In symbols, it is $\{(111)^n \mid n = 0, 1, 2, \dots\} = \{1^{3n} \mid n = 0, 1, 2, \dots\}$.

c) A little thought will show that this consists of all bit strings in which every 1 is immediately preceded by a 0. No other restrictions are imposed, since $0 \in A$.

d) Because the 0 appears only in 101, the strings formed here have the property that there are at least two 1's between every pair of 0's in the string, and the string begins and ends with a 1. All strings satisfying this property are in A^*.

7. This follows directly from the definition. Every string w in A^* consists of the concatenation of one or more strings from A. Since $A \subseteq B$, all of these strings are also in B, so w is the concatenation of one or more strings from B, i.e., is in B^*.

9. **a)** This set contains all bit strings, so of course the answer is yes.

b) This set contains all strings consisting of any number of 1's, followed by any number of 0's, followed by any number of 1's. Since 11101 is such a string, the answer is yes.

c) Any string belonging to this set must start 110, and 11101 does not, so the answer is no.

d) All the strings in this set must in particular have even length. The given string has odd length, so the answer is no.

e) The answer is yes. Just take one copy of each of the strings 111 and 0, together with the required string 1.

f) The answer is yes again. Just take 11 from the first set and 101 from the second.

11. In each case we will list the states in the order that they are visited, starting with the initial state. All we need to do then is to note whether the place we end up is a final state (s_0 or s_3) or a nonfinal state. (It is interesting to note that there are no transitions to s_3, so this state can never be reached.)

a) We encounter $s_0s_1s_2s_0$, so this string is accepted.

b) We encounter $s_0s_0s_0s_1s_2$, so this string is not accepted.

c) We encounter $s_0s_1s_0s_1s_0s_1s_2s_0$, so this string is accepted.

d) We encounter $s_0s_0s_1s_2s_0s_1s_2s_0s_1s_2$, so this string is not accepted.

13. **a)** The set in question is the set of all strings of zero or more 0's. Since the machine in Figure 1 has s_0 as a final state, and since there is a transition from s_0 to itself on input 0, every string of zero or more 0's will leave the machine in state s_0 and will therefore be accepted. Therefore the answer is yes.

b) Since this set is a subset of the set in part **(a)**, the answer must be yes.

c) One string in this set is the string 1. Since an input of 1 drives the machine to the nonfinal state s_1, not every string in this set is accepted. Therefore the answer is no.

d) One string in this set is the string 01. Since an input of 01 drives the machine to the nonfinal state s_1, not every string in this set is accepted. Therefore the answer is no.

e) The answer here is no for exactly the same reason as in part **(d)**.

f) The answer here is no for exactly the same reason as in part **(c)**.

15. We use structural induction on the input string y. The basis step is $y = \lambda$, and for the inductive step we write $y = wa$, where $w \in I^*$ and $a \in I$. For the basis step, we have $xy = x$, so we must show that $f(s,x) = f(f(s,x),\lambda)$. But part (i) of the definition of the extended transition function says that this is true. We then assume the inductive hypothesis that the equation holds for shorter strings and try to prove that $f(s,xwa) = f(f(s,x),wa)$. By part (ii) of the definition, the left-hand side of this equation equals $f(f(s,xw),a)$. By the inductive hypothesis (because w is shorter than y), $f(s,xw) = f(f(s,x),w)$, so $f(f(s,xw),a) = f(f(f(s,x),w),a)$. On the other hand, the right-hand side of our desired equality is, by part (ii) of the definition, equal to $f(f(f(s,x),w),a)$. We have shown that the two sides are equal, and our proof is complete.

17. The only final state is s_2, so we need to determine which strings drive the machine to state s_2. Clearly the strings 0, 10, and 11 do so, as well as any of these strings followed by anything else. Thus we can write the answer as $\{0,10,11\}\{0,1\}^*$.

19. A string is accepted if and only if it drives this machine to state s_1. Thus the string must consist of zero or more 0's, followed by a 1, followed by zero or more 1's. In short, the answer is $\{\, 0^m1^n \mid m \geq 0 \wedge n \geq 1 \,\}$.

21. Because s_0 is final, the empty string is accepted. The strings that drive the machine to final state s_3 are precisely $\{0\}\{1\}^*\{0\}$. There are three ways to get to final state s_4, and once we get there, we stay there. The path through s_2 tells us that strings in $\{10,11\}\{0,1\}^*$ are accepted. The path $s_0s_1s_3s_4$ tells us that strings in $\{0\}\{1\}^*\{01\}\{0,1\}^*$ are accepted. And the path $s_0s_1s_3s_5s_4$ tells us that strings in $\{0\}\{1\}^*\{00\}\{0\}^*\{1\}\{0,1\}^*$ are accepted. Thus the language recognized by this machine is $\{\lambda\}\cup\{0\}\{1\}^*\{0\}\cup\{10,11\}\{0,1\}^*\cup\{0\}\{1\}^*\{01\}\{0,1\}^*\cup\{0\}\{1\}^*\{00\}\{0\}^*\{1\}\{0,1\}^*$.

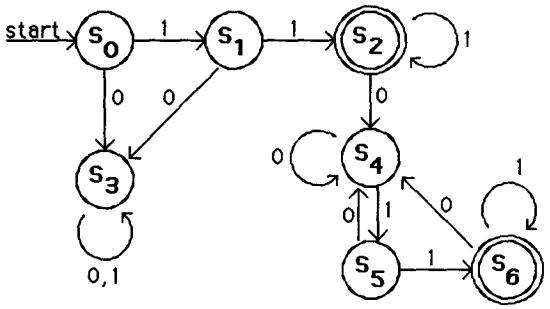
23. We want to accept only the strings that begin 01. Let s_2 be the only final state, and put transitions from s_2 to itself on either input. We want to reach s_2 after encountering 01, so put a transition from the start state s_0 to s_1 on input 0, and a transition from s_1 to s_2 on input 1. Finally make a "graveyard" state s_3, and have the other transitions from s_0 and s_1 (as well as both transitions from s_3) lead to s_3.

25. We can have a sequence of three states to record the appearance of 101. State s_1 will signify that we have just seen a 1; state s_2 will signify that we have just seen a 1 followed by a 0; state s_3 will be the only final state and will signify that we have seen the string 101. Put transitions from s_3 to itself on either input (it doesn't matter what follows the appearance of 101). Put a transition from the start state s_0 to itself on input 0, because we are still waiting for a 1. Put a transition from s_0 to s_1 on input 1 (because we have just seen a 1). From s_1 on input 0 we want to go to state s_2, but on input 1 we stay at s_1 because we have still just seen a 1. Finally, from s_2, put a transition on input 1 to the final state s_3 (success!), but on input 0 we have to start over looking for 101, so this transition must be back to s_0.

27. We can let state s_i, for $i = 0,1,2,3$ represent that exactly i 0's have been seen, and state s_4 will represent that four or more 0's have been seen. Only s_3 will be final. For $i = 0,1,2,3$, we transition from s_i to itself on input 1 and to s_{i+1} on input 0. Both transitions from s_4 are to itself.

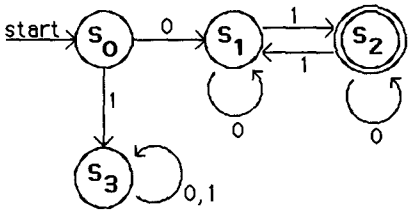
29. We can let state s_i, for $i = 0,1,2,3$ represent that i consecutive 1's have been seen. Only s_3 will be final. For $i = 0,1,2$, we transition from s_i to s_{i+1} on input 1 but back to s_0 on input 0. Both transitions from s_3 are to itself.

31. This is a little tricky. We want states at the start that prevent us from accepting a string if it does not start with 11. Once we have seen the first two 1's, we can accept the string if we do not encounter a 0 (after all, the strings 11 and 111 do satisfy the condition). We can also accept the string if it has anything whatsoever in the middle, as long as it ends 11. The machine shown below accomplishes all this. Note that s_3 is a graveyard state, and state s_4 is where we "start over" looking for the final 11.



33. We need just two states, s_0 to represent having seen an even number of 0's (this will be the start state, because to begin we have seen no 0's), and s_1 to represent having seen an odd number of 0's (this will be the only final state). The transitions are from each state to itself on input 1, and from each state to the other on input 0.

35. This is similar to Exercise 33, except that we need to look for the initial 0. Note that s_3 is the graveyard.



37. We prove this by contradiction. Suppose that such a machine exists, with start state s_0 and other state s_1. Because the empty string is not in the language but some strings are accepted, we must have s_1 as the only final state, with at least one transition from s_0 to s_1. Because the string 0 is not in the language, any transition from s_0 on input 0 must be to itself, so there must be a transition from s_0 to s_1 on input 1. But this cannot happen, because the string 1 is not in the language. Having obtained a contradiction, we conclude that no such finite-state automaton exists.

39. We want the new machine to accept exactly those strings that the original machine rejects, and vice versa. So we simply change each final state to a nonfinal state and change each nonfinal state to a final state.

41. We use exactly the same machine as in Exercise 25, but make s_0, s_1, and s_2 the final states and make s_3 nonfinal.

43. First some general comments on Exercises 43–49: In general it is quite hard to describe succinctly languages recognized by machines. An ad hoc approach is usually best. In this exercise there is only one final state, s_2, and only three ways to get there, namely on input 0, 01, or 11. Therefore the language recognized by this machine is $\{0, 01, 11\}$.

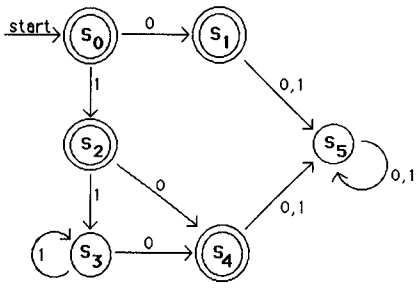
45. Clearly the empty string is accepted. There are essentially two ways to get to the final state s_2. We can go through state s_1, and every string of the form 0^n1^m, where n and m are positive integers, will take us through state s_1 on to s_2. We can also bypass state s_1, and every string of the form 01^m for $m \geq 0$ will take us directly to s_2. Thus our answer is $\{\lambda\} \cup \{\, 0^n1^m \mid n, m \geq 1 \,\} \cup \{\, 01^m \mid m \geq 0 \,\}$. Note that this can also be written as $\{\lambda, 0\} \cup \{\, 0^n1^m \mid n, m \geq 1 \,\}$.

47. First it is easy to see that all strings of the form 10^n for $n \geq 0$ can drive the machine to the final state s_1. Next we see that all strings of the form 10^n10^m for $n, m \geq 0$ can drive the machine to state s_3. No other strings can drive the machine to a final state. Therefore the answer is $\{\, 10^n \mid n \geq 0 \,\} \cup \{\, 10^n10^m \mid n, m \geq 0 \,\}$.

49. We notice first that state s_2 is a final state, that once we get there, we can stay there, and that any string that starts with a 0 can lead us there. Therefore all strings that start with a 0 are in the language. If the string starts with a 1, then we must go first to state s_1. If we ever leave state s_1, then the string will not be accepted, because there are no paths out of s_1 that lead to a final state. Therefore the only other strings that are in the language are the empty string (because s_0 is final) and those strings that can drive the machine to state s_1, namely strings consisting of all 1's (we've already included those of the form 01^*). Therefore the language accepted by this machine is the union of the set of all strings that start with a 0 and the set of all strings that have no 0's.

51. One way to do Exercises 50–54 is to construct a machine following the proof of Theorem 1. Rather than do that, we construct the machines in an ad hoc way, using the answers obtained in Exercises 43–47. Since λ, 0, and 1 are accepted by the nondeterministic automaton in Exercise 44, we make states s_0, s_1, and s_2 in the

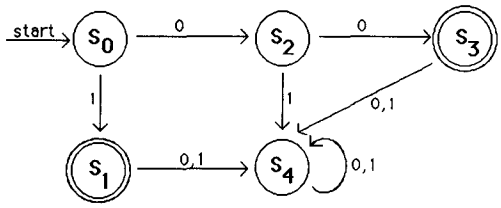
following diagram final. States s_3 and s_4 provide for the acceptance of strings of the form 1^n0 for all $n \geq 1$. State s_5, the graveyard state, assures that no other strings are accepted.



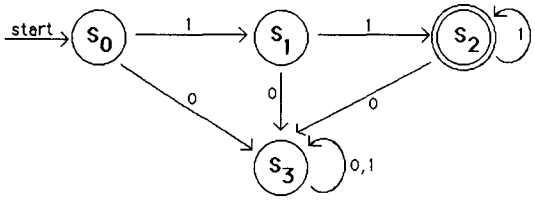
53. This machine is practically deterministic already, since there are no cases of ambiguous transitions (a given input allowing transition to more than one state). All that keeps this machine from being deterministic is that there are no transitions from certain states on certain inputs. Therefore to make this machine deterministic, we just need to add a "graveyard" state, s_3, with transitions from s_0 on input 0 and from s_1 on input 1 to this graveyard state, and transitions from s_3 to itself on input 0 or 1. The graveyard state is not final, of course.

55. **a)** We want to accept only the string 0. Let s_1 be the only final state, where we reach s_1 on input 0 from the start state s_0. Make a "graveyard" state s_2, and have all other transitions (there are five of them in all) lead there.

b) This uses the same idea as in part **(a)**, but we need a few more states. The graveyard state is s_4. See the picture for details.



c) In the picture of our machine, we show a transition to the graveyard state whenever we encounter a 0. The only final state is s_2, which we reach after 11 and remain at as long as the input consists just of 1's.



57. Intuitively, the reason that a finite-state automaton cannot recognize the set of bit strings containing an equal number of 0's and 1's is that there is not enough "memory" in the machine to keep track of how many extra 0's or 1's the machine has read so far. Of course, this intuition does not constitute a proof—maybe we are just not being clever enough to see how a machine could do this with a finite number of states. Instead, we must give a proof of this assertion. See Exercises 22–25 of Section 13.4 for a development of what are called "pumping lemmas" to handle various problems like this. (See also Example 6 in Section 13.4.)

The natural way to prove a negative statement such as this is by contradiction. So let us suppose that we do have a finite-state automaton M that accepts precisely the set of bit strings containing an equal number of 0's and 1's. We will derive a contradiction by showing that the machine must accept some illegal strings. The

idea behind the proof is that since there are only finitely many states, the machine must repeat some states as it computes. In this way, it can get into arbitrarily long loops, and this will lead us to a contradiction. To be specific, suppose that M has n states. Consider the string $0^{n+1}1^{n+1}$. As the machine processes this string, it must encounter the same state more than once as it reads the first $n+1$ 0's (by the pigeonhole principle). Say that it hits state s twice. Then some positive number, say k, of 0's in the input drives M from state s back to state s. But then the machine will end up at exactly the same place after reading $0^{n+1+k}1^{n+1}$ as it will after reading $0^{n+1}1^{n+1}$, since those extra k 0's simply drove it in a loop. Therefore since M accepts $0^{n+1}1^{n+1}$, it also accepts $0^{n+1+k}1^{n+1}$. But this is a contradiction, since this latter string does not have the same number of 0's as 1's.

59. We know from Exercise 58d that the equivalence classes of R_k are a refinement of the equivalence classes of R_{k-1} for each positive integer k. The equivalence classes are finite sets, and finite sets cannot be refined indefinitely (the most refined they can be is for each equivalence class to contain just one state). Therefore this sequence of refinements must stabilize and remain unchanged from some point onward. It remains to show that as soon as we have $R_n = R_{n+1}$, then $R_n = R_m$ for all $m > n$, from which it follows that $R_n = R_*$, and so the equivalence classes for these two relations will be the same. By induction, it suffices to show that if $R_n = R_{n+1}$, then $R_{n+1} = R_{n+2}$. By way of contradiction, suppose that $R_{n+1} \neq R_{n+2}$. This means that there are states s and t that are $(n+1)$-equivalent but not $(n+2)$-equivalent. Thus there is a string x of length $n+2$ such that, say, $f(s,x)$ is final but $f(t,x)$ is nonfinal. Write $x = aw$, where $a \in I$. Then $f(s,a)$ and $f(t,a)$ are not $(n+1)$-equivalent, because w drives the first to a final state and the second to a nonfinal state. But $f(s,a)$ and $f(t,a)$ are n-equivalent, because s and t are $(n+1)$-equivalent. This contradicts the fact that $R_n = R_{n+1}$, and our proof is complete.

61. **a)** By the way the machine $\overline{M}$ was constructed, a string will drive M from the start state to a final state if and only if that string drives $\overline{M}$ from the start state to a final state.

b) For a proof of this theorem, see a source such as *Introduction to Automata Theory, Languages, and Computation* (2nd Edition) by John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman (Addison Wesley, 2000).

SECTION 13.4 Language Recognition

Finding good verbal descriptions of the set of strings generated by a regular expression is not easy; neither is finding a good regular expression for a given verbal description. What Kleene's theorem says is that these problems of "programming" in regular expressions are really the same as the programming problems for machines discussed in the previous section. The **pumping lemma**, *discussed in Exercise 22 and the three exercises that follow it, is an important technique for proving that certain sets are not regular.*

1. **a)** This regular expression generates all strings consisting of zero or more 1's, followed by a lone 0.

b) This regular expression generates all strings consisting of zero or more 1's, followed by one or more 0's.

c) This set has only two elements, 111 and 001.

d) This set contains all strings in which the 0's come in pairs.

e) This set consists of all strings in which every 1 is preceded by at least one 0, with the proviso that the string ends in a 1 if it is not the empty string.

f) This gives us all strings of length at least 3 that end 00.

3. In each case we try to view 0101 as fitting the regular expression description.

a) The strings described by this regular expression have at most three "blocks" of different digits—a 0, then some 1's, then some 0's. Thus we cannot get the string 0101, which has four blocks.

b) The 1's that might come between the first and second 0 in any string described by this regular expression must come in pairs (because of the $(11)^*$). Therefore we cannot get 0101. Alternatively, note that every string described by this regular expression must have odd length.

c) We can get this string as $0(10)^1 1^1$.

d) We can get this string as $0^1 10(1)$, where the final 1 is one of the choices in $(0 \cup 1)$.

e) We can get this string as $(01)^2(11)^0$.

f) We cannot get this string, because every string with any 1's at all described by this regular expression must end with 10 or 11.

g) We cannot get this string, because every string described by this regular expression must end with 11.

h) We can get this string as $01(01)1^0$, where the second 01 is one of the choices in $(01 \cup 0)$.

5. **a)** We just need to take a union: $\mathbf{0} \cup \mathbf{11} \cup \mathbf{010}$.

b) More simply put, this is the set of strings of five or more 0's, so the regular expression is $\mathbf{000000^*}$.

c) We can use $(\mathbf{0} \cup \mathbf{1})$ to represent any symbol and $(\mathbf{00} \cup \mathbf{01} \cup \mathbf{10} \cup \mathbf{11})$ to represent any string of even length. We need one symbol followed by any string of even length, so we can take $(\mathbf{0} \cup \mathbf{1})(\mathbf{00} \cup \mathbf{01} \cup \mathbf{10} \cup \mathbf{11})^*$.

d) The one 1 can be preceded and/or followed by any number of 0's, so we have $\mathbf{0^*10^*}$.

e) This one is a little harder. In order to prevent 000 from appearing, we must have every group of one or two 0's followed by a 1 (if we note that the entire string ends with a 1 as well). Thus we can break our string down into groups of 1, 01, or 001, and we get $(\mathbf{1} \cup \mathbf{01} \cup \mathbf{001})^*$ as our regular expression.

7. **a)** We can translate "one or more 0's" into $\mathbf{00^*}$. Therefore the answer is $\mathbf{00^*1}$.

b) We can translate "two or more symbols" into $(\mathbf{0} \cup \mathbf{1})(\mathbf{0} \cup \mathbf{1})(\mathbf{0} \cup \mathbf{1})^*$. Therefore the answer is $(\mathbf{0} \cup \mathbf{1})(\mathbf{0} \cup \mathbf{1})(\mathbf{0} \cup \mathbf{1})^*\mathbf{0000^*}$.

c) A little thought tells us that we want all strings in which all the 0's come before all the 1's or all the 1's come before all the 0's. Thus the answer is $\mathbf{0^*1^*} \cup \mathbf{1^*0^*}$.

d) The string of 1's can be represented by $\mathbf{11(111)^*}$; the string of 0's, by $\mathbf{(00)^*}$. Thus the answer is $\mathbf{11(111)^*(00)^*}$.

9. **a)** The simplest solution here is to have just the start state s_0, nonfinal, with no transitions.

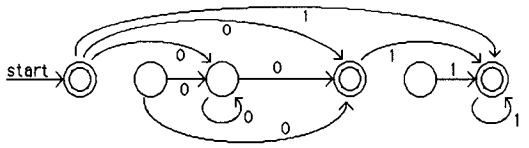
b) The simplest solution here is to have just the start state s_0, final, with no transitions.

c) The simplest solution here is to have just two states—the nonfinal start state s_0 (since we do not want to accept the empty string) and a final state s_1—and just the one transition from s_0 to s_1 on input a.

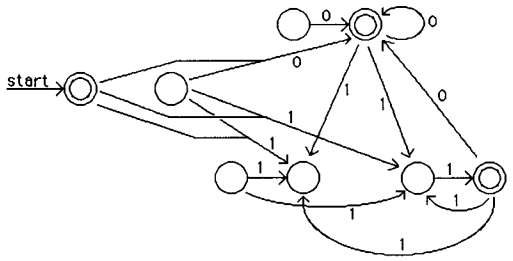
11. We can prove this by induction on the length of a regular expression for A. If this expression has length 1, then it is either $\emptyset$ or λ or x (where x is some symbol in the alphabet). In each case A is its own reversal, so there is nothing to prove. There are three inductive steps. If the regular expression for A is $\mathbf{BC}$, then $A = BC$, where B is the set generated by $\mathbf{B}$ and C is the set generated by $\mathbf{C}$. By the inductive hypothesis, we know that there are regular expressions $\mathbf{B'}$ and $\mathbf{C'}$ that generate B^R and C^R, respectively. Now $A^R = (BC)^R = (C^R)(B^R)$. Therefore a regular expression for A^R is $\mathbf{C'B'}$. The case of union is handled similarly. Let the regular expression for A be $\mathbf{B} \cup \mathbf{C}$, with B, C, $\mathbf{B'}$, and $\mathbf{C'}$ as before. Then a regular expression for A^R is $\mathbf{B'} \cup \mathbf{C'}$, since clearly $(B \cup C)^R = (B^R) \cup (C^R)$. Finally, if the regular expression for A is $\mathbf{B^*}$, then, with the same notation as before, it is easy to see that $(\mathbf{B'})^*$ is a regular expression for A^R.

13. **a)** We can build machines to recognize $\mathbf{0^*}$ and $\mathbf{1^*}$ as shown in the second row of Figure 3. Next we need to put these together to make a machine that recognizes $\mathbf{0^*1^*}$. We place the first machine on the left and

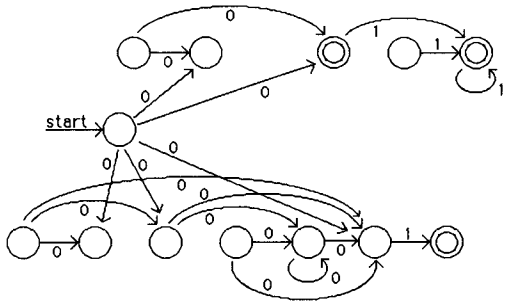
the second machine on the right. We make each final state in the first machine nonfinal (except for the start state, since $\lambda \in \mathbf{0^*1^*}$), but leave the final states in the second machine final. Next we copy each transition to a state that was formerly final in the first machine into a transition (on the same input) to the start state of the second machine. Lastly, since $\lambda \in \mathbf{0^*}$, we add the transition from the start state to the state to which there is a transition from the start state of the machine for $\mathbf{1^*}$. The result is as shown. (In all parts of this exercise we have not put names on the states in our state diagrams.)



b) This machine is quite messy. The upper portion is for $\mathbf{0}$, and the lower portion is for $\mathbf{11}$. They are combined to give a machine for $\mathbf{0 \cup 11}$. Finally, to incorporate the Kleene star, we added a new start state (on the far left), and adjusted the transitions according to the procedure shown in Figure 2.



c) This is similar to the other parts. We grouped the expression as $\mathbf{01^* \cup (00^*)1}$. The answer is as shown.



15. We choose as the nonterminal symbols corresponding to states s_0, s_1, and s_2 the symbols S, A, and B, respectively. Thus S is our start symbol. The terminal symbols are of course 0 and 1. We construct the rules for our grammars by following the procedure described in the proof of the second half of Theorem 2: putting in rules of the form $X \to aY$ for each transition from the state corresponding to X to the state corresponding to Y, on input a, and putting in a rule of the form $X \to a$ for a transition from the state corresponding to X to the final state, on input a. Specifically, since there is a transition from s_0 to s_1 on input 0, we include the rule $S \to 0A$. Similarly, the other transitions give us the rules $S \to 1B$, $A \to 0B$, $A \to 1B$, $B \to 0B$, and $B \to 1B$. Also, the transition to the final state from S on input 0 gives rise to the rule $S \to 0$. Thus our grammar contains these seven rules.

17. This is similar to Exercise 15—see the discussion there for the approach. We let C correspond to state s_3. The set of rules contains $S \to 0C$, $S \to 1A$, $A \to 1A$, $A \to 0C$, $B \to 0B$, $B \to 1B$, $C \to 0C$, and $C \to 1B$ (for the transitions from a state to another state on a given input), as well as $S \to 1$, $A \to 1$, $B \to 0$, $B \to 1$, and $C \to 1$ (for the transitions to the final states that can end the computation).

19. This is clear, since the operation of the machine is exactly mimicked by the grammar. If the current string in the derivation in the grammar is $v_1v_2 \ldots v_kA_s$, then the machine has seen input $v_1v_2 \ldots v_k$ and is currently in state s. If the current string in the derivation in the grammar is $v_1v_2 \ldots v_k$, then the machine has seen input $v_1v_2 \ldots v_k$ and is currently in some final state. Hence the machine accepts precisely those strings that the grammar generates. (The empty string does not fit this discussion, but it is handled separately—and correctly—since we take $S \rightarrow \lambda$ as a production if and only if we are supposed to.)

21. First suppose that the language recognized by M is infinite. Then the length of the words recognized by M must be unbounded, since there are only a finite number of symbols. Thus $l(x)$ is greater than the finite number $|S|$ for some word $x \in L(M)$.

Conversely, let x be such a word, and let $s_0, s_{i_1}, s_{i_2}, \ldots, s_{i_n}$ be the sequence of states that the machine goes through on input x, where $n = l(x)$ and s_{i_n} is a final state. By the pigeonhole principle, some state occurs twice in this sequence, i.e., there is a loop from this state back to itself during the computation. Let y be the substring of x that causes the loop, so that $x = uyv$. Then for every nonnegative integer k, the string uy^kv is accepted by the machine M (i.e., is in $L(M)$), since the computation is the same as the computation on input x, except that the loop is traversed k times. Thus $L(M)$ is infinite.

23. We apply the pumping lemma in a proof by contradiction. Suppose that this set were regular. Clearly it contains arbitrarily long strings. Thus the pumping lemma tells us that for some strings u, $v \neq \lambda$, and w, the string uv^iw is in our set for every i. Now if v contains both 0's and 1's, then uv^2w cannot be in the set, since it would have a 0 following a 1, which no string in our set has. On the other hand, if v contains only 0's (or only 1's), then for large enough i, it is clear that uv^iw has more than (or less than) twice as many 0's as 1's, again contradicting the definition of our set. Thus the set cannot be regular.

25. We will give a proof by contradiction, using the pumping lemma. Following the hint, let x be the palindrome 0^N10^N, for some fixed $N > |S|$, where S is the set of states in a machine that recognizes palindromes. By the lemma, we can write $x = uvw$, with $l(uv) \leq |S|$ and $l(v) \geq 1$, so that for all i, uv^iw is a palindrome. Now since $0^N10^N = uvw$ and $l(uv) \leq |S| < N$, it must be the case that v is a string consisting solely of 0's, with the 1 lying in w. Then uv^2w cannot be a palindrome, since it has more 0's before its sole 1 than it has 0's following the 1.

27. It helps to think of L/x in words—it is the set of "ends" of strings in L that start with the string x; in other words, it is the set of strings obtained from strings in L by stripping away an initial piece x. To show that 11 and 10 are distinguishable, we need to find a string z such that $11z \in L$ and $10z \notin L$ or vice versa. A little thought and trial and error shows us that $z = 1$ works: $111 \notin L$ but $101 \in L$. To see that 1 and 11 are indistinguishable, note that the only way for $1z$ to be in L is for z to end with 01, and that is also the only way for $11z$ to be in L.

29. By Exercise 28, if two strings are distinguishable, then they drive the machine from the start state to different states. Therefore, if $x_1, x_2, \ldots, x_n$ are all distinguishable, the states $f(s_0, x_1), f(s_0, x_2), \ldots, f(s_0, x_n)$ are all different, so the the machine has at least n states.

31. We claim that any two distinct strings of the same length are distinguishable with respect to the language P of all palindromes. Indeed, if x and y are distinct strings of length n, let $z = x^R$ (the reverse of string x). Then $xz \in P$ but $yz \notin P$. Note that there are 2^n different strings of length n. By Exercise 29, this tells us that any deterministic finite-state automaton for recognizing palindromes must have at least 2^n states. Because n is arbitrary (we want our machine to recognize *all* palindromes), this tells us that no finite-state machine can recognize P.

SECTION 13.5 Turing Machines

In this final section of the textbook, we have studied a machine that has all the computing capabilities possible (if one believes the Church–Turing thesis). Most of these exercises are really programming assignments, and the programming language you are stuck with is not a nice, high-level, structured language like Java or C, nor even a nice assembly language, but something much messier and less efficient. One point of the exercises is to convince you that even in this horrible setting you can, with enough time and patience, instruct the computer—the Turing machine—to do whatever you wish computationally. Keep in mind that in many senses, a Turing machine is just as powerful as any computer running programs written in any language. One reason for talking about Turing machines at all, rather than just using high-level languages, is that their simplicity makes it feasible to prove some very interesting things about them (and therefore about computers in general). For example, one can prove that computers cannot solve the halting problem (see also Section 3.1), and one can prove that a large class of problems have efficient algorithmic solutions if and only if certain very specific problems, such as a decision version of the traveling salesman problem, do (the NP-complete problems—see also Section 3.3). This is part of what makes Turing machines so important in theoretical computer science, and time spent becoming acquainted with them will not go unrewarded as you progress in this field.

1. We will indicate the configuration of the Turing machine using a notation such as $0[s_2]1B1$. This string of symbols means that the tape is blank except for a portion which reads $01B1$ from left to right; that the machine is currently in state s_2; and that the tape head is reading the left 1 (the currently scanned symbol will always be the one following the bracketed state information).

a) The initial configuration is $[s_0]0011$. Because of the five-tuple $(s_0, 0, s_1, 1, R)$ and the fact that the machine is in state s_0 and the tape head is looking at a 0, the machine changes the 0 to a 1 (i.e., writes a 1 in that square), moves to the right, and enters state s_1. Therefore the configuration at the end of one step of the computation is $1[s_1]011$. Next the transition given by the five-tuple $(s_1, 0, s_2, 1, L)$ occurs, and we reach the configuration $[s_2]1111$. There are no five-tuples starting with s_2, so the machine halts at this point. The nonblank portion of the tape contains 1111.

b) The initial configuration is $[s_0]101$. Because of the five-tuple $(s_0, 1, s_1, 0, R)$ and the fact that the machine is in state s_0 and the tape head is looking at a 1, the machine changes the 1 to a 0, moves to the right, and enters state s_1. Therefore the configuration at the end of one step of the computation is $0[s_1]01$. At this time transition $(s_1, 0, s_2, 1, L)$ kicks in, resulting in configuration $[s_2]011$, and the machine halts, with 011 on its tape.

c) We seem to have the idea from the first two parts, so let us just list the configurations here, using the notation "$\rightarrow$" to show the progression from one to the next. $[s_0]11B01 \rightarrow 0[s_1]1B01 \rightarrow 00[s_1]B01 \rightarrow 0[s_2]0001$. Therefore the final output is 00001.

d) $[s_0]B \rightarrow 0[s_1]B \rightarrow [s_2]00$. So the final tape reads 00.

3. Note that all motion is from left to right.

a) The machine starts in state s_0 and sees the first 1. Therefore using the second five-tuple, it replaces the 1 by a 0, moves to the right, and enters state s_1. Now it sees the second 1, so, using the fifth five-tuple, it replaces the 1 by a 1 (i.e., leaves it unchanged), moves to the right, and enters state s_0. The third five-tuple now tells it to leave the blank it sees alone, move to the right, and enter state s_2, which is a final (accepting) state (because it is not the first state in any five-tuple). Since there are no five-tuples telling the machine what to do in state s_2, it halts. Note that 01 is on the tape, and the input was accepted.

b) When in state s_0 the machine skips over 0's, ignoring them, until it comes to a 1. When (and if) this happens, the machine changes this 1 to a 0 and enters state s_1. Note also that if the machine hits a blank (B) while in state s_0 or s_1, then it enters the final (accepting) state s_2. Next note that s_1 plays a role similar to that played by s_0, causing the machine to skip over 0's, but causing it to go back into state s_0 if and when it encounters a 1. In state s_1, however, the machine does not change the 1 it sees to a 0. Thus

the machine will alternate between states s_0 and s_1 as it encounters 1's in the input string, changing half of these 1's to 0's. To summarize, if the machine is given a bit string as input, it scans it from left to right, changing every other occurrence of a 1, if any, starting with the first, to a 0, and otherwise leaving the string unchanged; it halts (and accepts) when it comes to the end of the string.

5. **a)** The machine starts in state s_0 and sees the first 1. Therefore using the first five-tuple, it replaces the 1 by a 0, moves to the right, and enters state s_1. Now it sees the second 1, so, using the second five-tuple, it replaces the 1 by a 1 (i.e., leaves it unchanged), moves to the right, and stays in state s_1. Since there are no five-tuples telling the machine what to do in state s_1 when reading a blank, it halts. Note that 01 is on the tape, and the input was not accepted, because s_1 is not a final state; in fact, there are no final states (states that begin no 5-tuples).
b) This is essentially the same as part **(a)**. The first 1 (if any) is changed to a 0 and the others are left alone. The input is not accepted.

7. The machine needs to search for the first 0 and when (and if) it finds it, replace it with a 1. So let's have the machine stay in its initial state (s_0) as long as it reads 1's, constantly moving to the right. If it ever reads a 0 it will enter state s_1 while changing the 0 to a 1. No further action is required. Thus we can get by with just the following two five-tuples: $(s_0, 0, s_1, 1, R)$ and $(s_0, 1, s_0, 1, R)$. Note that if the input string consists of just 1's, then the machine eventually sees the terminating blank and halts.

9. The machine should scan the tape, leaving it alone until it has encountered the first 1. At that point, it needs to enter a phase in which it changes all the 1's to 0's, until it reaches the end of the input. So we'll have tuples $(s_0, 0, s_0, 0, R)$ and $(s_0, 1, s_1, 1, R)$ to complete the first phase, and then have tuples $(s_1, 0, s_1, 0, R)$ and $(s_1, 1, s_1, 0, R)$ to complete the second phase. When the machine encounters the end of the input (a blank on the tape) it halts, since there are no transitions given with a blank as the scanned symbol.

11. We can have the machine scan the input tape until it reaches the first blank, "remembering" what the last symbol was that it read. Let us use state s_0 to represent that last symbol's being a 1, and s_1 to represent its being a 0. It doesn't matter what gets written, so we'll just leave the tape unchanged as we move from left to right. Thus our first few five-tuples are $(s_0, 0, s_1, 0, R)$, $(s_0, 1, s_0, 1, R)$, $(s_1, 0, s_1, 0, R)$, $(s_1, 1, s_0, 1, R)$. Now suppose the machine encounters the end of the input, namely the blank at the end of the input string. If it is in state s_0, then the last symbol read was not a 0, so we want to not accept the string. If it is in state s_1, then the last symbol read was a 0, so we want to accept the string. Recall that the convention presented in this section was that acceptance is indicated by halting in a final state, i.e., one with no transitions out of it. So let's add the five-tuple (s_1, B, s_2, B, R) for accepting when we should. To make sure we don't accept when we shouldn't, we need do nothing else, because the machine will halt in the nonfinal state s_0 in this case.

An alternative approach to this problem is to have the machine scan to the right until it reaches the end of the tape, then back up, "look" at the last symbol, and take the appropriate action.

13. This is very similar to Exercise 11. We want the machine to "remember" whether it has seen an even number of 1's or not. We'll let s_0 be the state representing that an even number of 1's have been seen (which is of course true at the start of the computation), and let s_1 be the state representing that an odd number of 1's have been seen. So we put in the following tuples: $(s_0, 0, s_0, 0, R)$, $(s_0, 1, s_1, 1, R)$, $(s_1, 0, s_1, 0, R)$, and $(s_1, 1, s_0, 1, R)$. When the machine encounters the terminating blank, we want it to accept if it is in state s_0, so we add the tuple (s_0, B, s_2, B, R). Thus the machine will halt in final state s_2 if the input string has an even number of 0's, and it will halt in nonfinal state s_1 otherwise.

15. You need to play with this machine to get a feel for what is going on. After doing so, you will understand that it operates as follows. If the input string is blank or starts with a 1, then the machine halts in state s_0, which is not final, and therefore every such string is not accepted (which is a good thing, since it is not in the set to be recognized). Otherwise the initial 0 is changed to an M, and the machine skips past all the intervening 0's and 1's until it either comes to the end of the input string or else comes to an M (which, as we will see, has been written over the right-most remaining bit). At this point it backs up (moves left) one square and is in state s_2. Since the acceptable strings must have a 1 at the right for each 0 at the left, there had better be a 1 here if the string is acceptable. Therefore the only transition out of state s_2 occurs when this square contains a 1. If it does, then the machine replaces it with an M, and makes its way back to the left. (If this square does not contain a 1, then the machine halts in the nonfinal state s_2, as appropriate.) On its way back, it stays in state s_3 as long as it sees 1's, then stays in s_4 as long as it sees 0's. Eventually either it encounters a 1 while in state s_4, at which point it (appropriately) halts without accepting (since the string had a 0 to the right of a 1); or else it reaches the right-most M that had been written over a 0 near the beginning of the string. If it is in state s_3 when this happens, then there are no more 0's in the string, so it had better be the case (if we want to accept this string) that there are no more 1's either; this is accomplished by the transitions (s_3, M, s_5, M, R) and (s_5, M, s_6, M, R), and s_6 is a final state. Otherwise, the machine halts in nonfinal state s_5. If it is in state s_4 when this M is encountered, then we need to start all over again, except that now the string will have had its left-most remaining 0 and its right-most remaining 1 replaced by M's. So the machine moves (staying in state s_4) to the left-most remaining 0 and goes back into state s_0 to repeat the process.

17. This will be similar to the machine in Example 3, in that we will change the digits one at a time to a new symbol M. We can't work from the outside in as we did there, however, so we'll replace all three digits from left to right. Furthermore, we'll put a new symbol, E, at the left end of the input in order to tell more easily when we have arrived back at the starting point. Here is our plan for the states and the transitions that will accomplish our goal. State s_9 is our (accepting) final state. States s_0 and s_1 will write an E to the left of the initial input and return to the first input square, entering state s_2. (If, however, the tape is blank, then the machine will accept immediately, and if the first symbol is not a 0, then it will reject immediately.) The five-tuples are (s_0, B, s_9, B, L), $(s_0, 0, s_1, 0, L)$, and (s_1, B, s_2, E, R). State s_2 will skip past any M's until it finds the first 0, change it to an M, and enter state s_3. The transitions are (s_2, M, s_2, M, R) and $(s_2, 0, s_3, M, R)$. Similarly, state s_3 will skip past any remaining 0's and any M's until it finds the first 1, change it to an M, and enter state s_4. The transitions are $(s_3, 0, s_3, 0, R)$, (s_3, M, s_3, M, R) and $(s_3, 1, s_4, M, R)$. State s_4 will do the same for the first 2 (skipping past remaining 1's and M's, and ending in state s_5), with transitions $(s_4, 1, s_4, 1, R)$, (s_4, M, s_4, M, R) and $(s_4, 2, s_5, M, R)$. State s_5 then will skip over any remaining 2's and (if there is any chance of accepting this string) encounter the terminating blank. The transitions are $(s_5, 2, s_5, 2, R)$ and (s_5, B, s_6, B, L). Note that once this blank has been seen, we back up to the last symbol before it and enter state s_6. There are now two possibilities. If the scanned square is an M, then we should accept if and only if the entire string consists of M's at this point. We will enter state s_8 to check this, with the transition (s_6, M, s_8, M, L). Otherwise, there will be a 2 here, and we want to go back to the start of the string to begin the cycle all over; we'll use state s_7 to accomplish this, so we put in the five-tuple $(s_6, 2, s_7, 2, L)$. In this latter case, the machine should skip over everything until it sees the marker E that we put at the left end of the input, then move back to the initial input square, and start over in state s_2. The transitions $(s_7, 0, s_7, 0, L)$, $(s_7, 1, s_7, 1, L)$, $(s_7, 2, s_7, 2, L)$, (s_7, M, s_7, M, L), and (s_7, E, s_2, E, R) accomplish this. But if we entered state s_8, then we need to make sure that there is nothing but M's all the way back to the starting point; we add the five-tuples (s_8, M, s_8, M, L) and (s_8, E, s_9, E, L), and we're finished.

19. Recall that functions are computed in a funny way using unary notation. The string representing n is a string of $n + 1$ 1's. Thus we want our machine to erase three of these 1's (or all but one of them, if there are

fewer than four), and then halt. One way to accomplish this is as follows. If $n \geq 3$, then the five-tuples $(s_0, 1, s_1, B, R)$, $(s_1, 1, s_2, B, R)$, $(s_2, 1, s_3, B, R)$, and $(s_3, 1, s_4, 1, R)$ will do the trick (s_4 is just a halting state). To account for the possibilities that $n < 3$, we add transitions $(s_1, B, s_4, 1, R)$, $(s_2, B, s_4, 1, R)$, and $(s_3, B, s_4, 1, R)$. In each of these three cases, we needed to restore one 1 before halting (since the "answer" was to be 0).

21. The machine here first needs to "decide" whether $n \geq 5$. If it finds that $n \geq 5$, then it needs to leave exactly four 1's on the tape (according to our rules for representing numbers in unary); otherwise it needs to leave exactly one 1. We'll use states s_0 through s_6 for this task, with the following five-tuples, which erase the tape as they move from left to right through the input: $(s_0, 1, s_1, B, R)$, $(s_1, 1, s_2, B, R)$, (s_1, B, s_6, B, R), $(s_2, 1, s_3, B, R)$, (s_2, B, s_6, B, R), $(s_3, 1, s_4, B, R)$, (s_3, B, s_6, B, R), $(s_4, 1, s_5, B, R)$, (s_4, B, s_6, B, R). At this point, the machine is either in state s_5 (and $n \geq 5$), or in state s_6 with a blank tape (and $n < 5$). To finish in the latter case, we just write a 1 and halt: $(s_6, B, s_{10}, 1, R)$. For the former case, we erase the rest of the tape, write four 1's, and halt: $(s_5, 1, s_5, B, R)$, $(s_5, B, s_7, 1, R)$, $(s_7, B, s_8, 1, R)$, $(s_8, B, s_9, 1, R)$, and $(s_9, B, s_{10}, 1, R)$.

23. We start with a string of $n + 1$ 1's, and we want to end up with a string of $3n + 1$ 1's. Our idea will be to replace the last 1 with a 0, then for each 1 to the left of the 0, write a pair of new 1's to the right of the 0. To keep track of which 1's we have processed so far, we will change each left-side 1 to a 0 as we process it. At the end, we will change all the 0's back to 1's. Basically our states will mean the following ("first" means "first encountered"): s_0, scan right for last 1; s_1, change the last 1 to 0; s_2, scan left to first 1; s_3, scan right for end of input (having replaced the 1 where we started with a 0); s_3 and s_4, write the two more 1's; s_5, scan left to first 0; s_6, replace the remaining 0's with 1's; s_7, halt.

The needed five-tuples are as follows: $(s_0, 1, s_0, 1, R)$, (s_0, B, s_1, B, L), $(s_1, 1, s_2, 0, L)$, $(s_2, 0, s_2, 0, L)$, $(s_2, 1, s_3, 0, R)$, (s_2, B, s_6, B, R), $(s_3, 0, s_3, 0, R)$, $(s_3, 1, s_3, 1, R)$, $(s_3, B, s_4, 1, R)$, $(s_4, B, s_5, 1, L)$, $(s_5, 1, s_5, 1, L)$, $(s_5, 0, s_2, 0, L)$, $(s_6, 0, s_6, 1, R)$, $(s_6, 1, s_7, 1, R)$, (s_6, B, s_7, B, R).

25. The idea here is to match off the 1's in the two inputs (changing the 1's to 0's from the left, say, to keep track), until one of them is exhausted. At that point, we need to erase the larger input entirely (as well as the asterisk) and change the 0's back to 1's. Here is how we'll do it. In state s_0 we skip over any 0's until we come to either a 1 or the $*$. If it's the $*$, then we know that the second input (n_2) is at least as large as the first (n_1), so we enter a clean-up state s_5, which erases the asterisk and all the 0's and 1's to its right. The five-tuples for this much are $(s_0, 0, s_0, 0, R)$, $(s_0, *, s_5, B, R)$, $(s_5, 1, s_5, B, R)$, and $(s_5, 0, s_5, B, R)$. Once this erasing is finished, we need to go over to the part of the tape where the first input was and change all the 0's back to 1's; the following transitions accomplish this: (s_5, B, s_6, B, L), (s_6, B, s_6, B, L), $(s_6, 0, s_7, 1, L)$, and $(s_7, 0, s_7, 1, L)$. Eventually the machine halts in state s_7 when the blank preceding the original input is encountered.

The other possibility is that the machine encounters a 1 while in state s_0. We want to change this 1 to a 0, skip over any remaining 1's as well as the asterisk, skip over any 0's to the right of the asterisk (these represent parts of n_2 that have already been matched off against equal parts of n_1), and then either find a 1 in n_2 (which we change to a 0) or else come to the blank at the end of the input. Here are the transitions: $(s_0, 1, s_1, 0, R)$, $(s_1, 1, s_1, 1, R)$, $(s_1, *, s_2, *, R)$, $(s_2, 0, s_2, 0, R)$, $(s_2, 1, s_3, 0, L)$, (s_2, B, s_4, B, L). At this point we are either in state s_3, ready to go back for the next iteration, or in state s_4 ready for some cleanup. In the former case, we want to skip back over the nonblank symbols until we reach the start of the string, so we add five-tuples $(s_3, *, s_3, *, L)$, $(s_3, 0, s_3, 0, L)$, $(s_3, 1, s_3, 1, L)$, and (s_3, B, s_0, B, R). In the latter case, we know that the first string is longer than the second. Therefore we want to change the 0's in the second input string back to 1's and then erase the asterisk and remnants of the first input string. Here are the transitions: $(s_4, 0, s_4, 1, L)$, $(s_4, *, s_8, B, L)$, $(s_8, 0, s_8, B, L)$, $(s_8, 1, s_8, B, L)$.

27. The discussion in the preamble tells how to take the machines from Exercises 22 and 18 and create a new machine. The only catch is that the tape head needs to be back at the leftmost 1. Suppose that s_m, where m is the largest index, is the state in which the Turing machine for Exercise 22 halts after completing its work, and suppose that we have designed that machine so that when the machine halts the tape head is reading the leftmost 1 of the answer. Then we renumber each state in the machine for Exercise 18 by adding m to each subscript, and take the union of the two sets of five-tuples.

29. If the answer is yes/no, then the problem is a decision problem.

a) No, the answer here is a number, not yes or no.

b) Yes, the answer is either yes or no.

c) Yes, the answer is either yes or no.

d) Yes, the answer is either yes or no.

31. This is a fairly hard problem, which can be solved by patiently trying various combinations. The following five-tuples will do the trick: $(s_0, B, s_1, 1, L)$, $(s_0, 1, s_1, 1, R)$, $(s_1, B, s_0, 1, R)$.

GUIDE TO REVIEW QUESTIONS FOR CHAPTER 13

1. **a)** See p. 849. **b)** See p. 849.

2. **a)** See p. 850. **b)** $\{0^{3n}1 \mid n \geq 0\}$

c) The vocabulary is $\{S, 0, 1\}$; the terminals are $T = \{0, 1\}$; the start symbol is S; and the productions are $S \to S1$ and $S \to 0$.

3. **a)** See p. 851. **b)** a grammar that contains a production like $AB \to C$

c) See p. 851. **d)** a grammar that contains a production like $Sa \to Sbc$

e) See p. 851. **f)** a grammar that contains a production like $S \to SS$

4. **a)** See p. 851. **b)** See p. 851.

c) See Example 8 in Section 13.1.

5. **a)** See p. 854. **b)** See Example 14 in Section 13.1.

6. **a)** See p. 851 (machines with output) and p. 867 (machines without output, called finite-state automata). See also p. 863 for comments on other types of finite-state machines.

b) Have three states and only one input symbol, Q (quarter). The start state s_0 has a transition to state s_1 on input Q and outputs nothing; state s_1 has a transition to state s_2 on input Q and outputs nothing; state s_2 has a transition back to state s_1 on input Q and outputs a drink.

7. $1^* \cup 1^*00$

8. Have four states, with only s_2 final. From the start state s_0, go to a graveyard state s_1 on input 0, and go to state s_2 on input 1. From both states s_2 and s_3, go to s_2 on input 1 and to s_3 on input 0.

9. **a)** See p. 866.

b) the set of all strings in which all the maximal blocks of consecutive 1's (if any) have an even number of 1's

10. **a)** See p. 867. **b)** See p. 868.

11. **a)** See p. 873. **b)** See Theorem 1 in Section 13.3.

12. **a)** See p. 879. **b)** See p. 879.