SECTION 11.1 Introduction to Trees

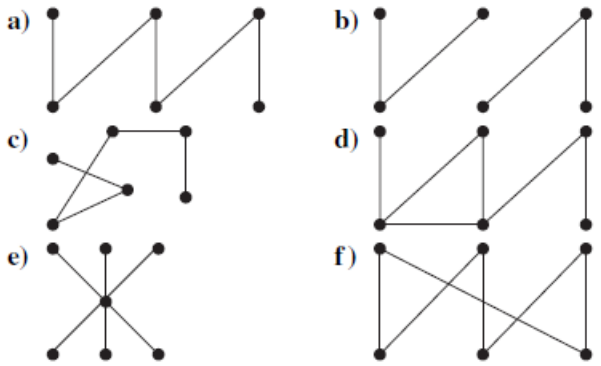
These exercises give the reader experience working with tree terminology, and in particular with the relationships between the height and the numbers of vertices, leaves, and internal vertices of a tree. Exercise 13 should be done to get a feeling for the structure of trees. One good way to organize 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.

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. Which of these graphs are trees?



---

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.

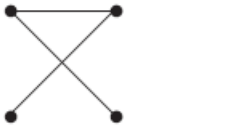
d) This graph has a simple circuit, so it is not a tree.

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.

2. Which of these graphs are trees?

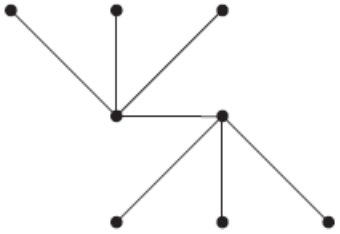
a)



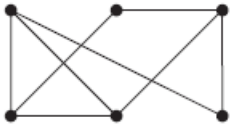
b)

c)

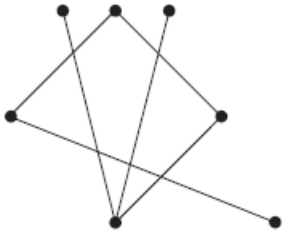
d)



e)



f)



2. a) This is a tree since it is connected and has no simple circuits.

b) This is a tree since it is connected and has no simple circuits.

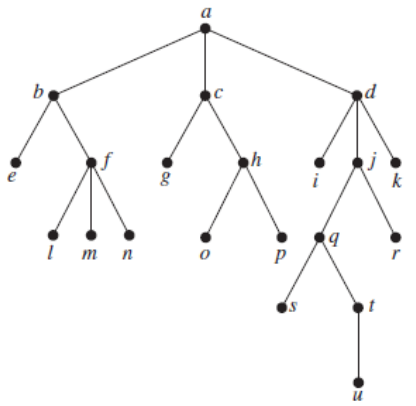
c) This is not a tree, since it is not connected.

d) This is a tree since it is connected and has no simple circuits.

e) This is not a tree, since it has a simple circuit.

f) This is a tree since it is connected and has no simple circuits.

3. Answer these questions about the rooted tree illustrated.



a) Which vertex is the root?
b) Which vertices are internal?
c) Which vertices are leaves?
d) Which vertices are children of j?
e) Which vertex is the parent of h?
f) Which vertices are siblings of o?
g) Which vertices are ancestors of m?
h) Which vertices are descendants of b?

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.

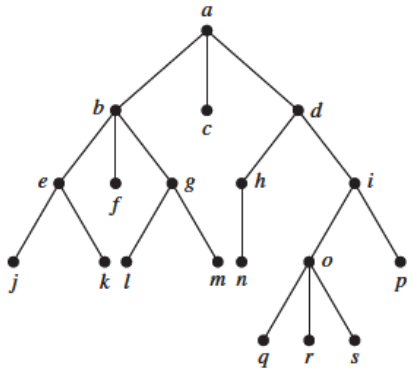
f) Vertex o has only one sibling, namely p, which is the other child of o's parent, h.

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.

---

4. Answer the same questions as listed in Exercise 3 for the rooted tree illustrated.



4. 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, d, e, g, h, i, and o.

c) The leaves are the vertices without children, namely c, f, j, k, l, m, n, p, q, r, and s.

d) The children of j are the vertices adjacent to j and below j. There are no such vertices, so there are no children.

e) The parent of h is the vertex adjacent to h and above h, namely d.

f) Vertex o has only one sibling, namely p, which is the other child of o's parent, i.

g) The ancestors of m are all the vertices on the unique simple path from m back to the root, namely g, b, and a.

h) The descendants of b are all the vertices that have b as an ancestor, namely e, f, g, j, k, l, and m.

---

5. Is the rooted tree in Exercise 3 a full m-ary tree for some positive integer m?

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.

---

6. Is the rooted tree in Exercise 4 a full m-ary tree for some positive integer m?

6. 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. What is the level of each vertex of the rooted tree in Exercise 3?

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.

---

8. What is the level of each vertex of the rooted tree in Exercise 4?

8. 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 i (in alphabetical order), are at level 2. Similarly j through p are at level 3, and q, r, and s are at level 4.

---

9. Draw the subtree of the tree in Exercise 3 that is rooted at

a) a. **b)** c. **c)** e.

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.

10. Draw the subtree of the tree in Exercise 4 that is rooted at

a) a. **b)** c. **c)** e.

10. 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 just the vertex c.

c) The subtree rooted at e consists of e, j, and k, and the edges ej and ek.

---

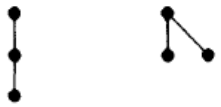
11. **a)** How many nonisomorphic unrooted trees are there with three vertices?

b) How many nonisomorphic rooted trees are there with three vertices (using isomorphism for directed graphs)?

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.



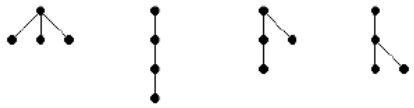
***12.** **a)** How many nonisomorphic unrooted trees are there with four vertices?

b) How many nonisomorphic rooted trees are there with four vertices (using isomorphism for directed graphs)?

12. 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 are two trees with four vertices, namely $K_{1,3}$ and the simple path of length 3. See the first two trees below.

b) The longest path from the root can have length 1, 2 or 3. There is only one tree with longest path of length 1 (the other three vertices are at level 1), and only one with longest path of length 3. If the longest path has length 2, then the fourth vertex (after using three vertices to draw this path) can be "attached" to either the root or the vertex at level 1, giving us two nonisomorphic trees. Thus there are a total of four nonisomorphic rooted trees on 4 vertices, as shown below.



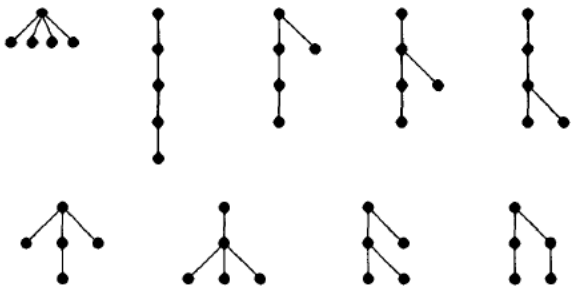
***13.** **a)** How many nonisomorphic unrooted trees are there with five vertices?

b) How many nonisomorphic rooted trees are there with five vertices (using isomorphism for directed graphs)?

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.



***14.** Show that a simple graph is a tree if and only if it is connected but the deletion of any of its edges produces a graph that is not connected.

14. There are two things to prove. First suppose that T is a tree. By definition it is connected, so we need to show that the deletion of any of its edges produces a graph that is not connected. Let $\{x, y\}$ be an edge of T, and note that $x \neq y$. Now T with $\{x, y\}$ deleted has no path from x to y, since there was only one simple path from x to y in T, and the edge itself was it. (We use Theorem 1 here, as well as the fact that if there is a path from a vertex u to another vertex v, then there is a simple path from u to v by Theorem 1 in Section 10.4.) Therefore the graph with $\{x, y\}$ deleted is not connected.

Conversely, suppose that a simple connected graph T satisfies the condition that the removal of any edge will disconnect it. We must show that T is a tree. If not, then T has a simple circuit, say $x_1, x_2, \ldots, x_r, x_1$. If we delete edge $\{x_r, x_1\}$ from T, then the graph will remain connected, since wherever the deleted edge was used in forming paths between vertices we can instead use the rest of the circuit: $x_1, x_2, \ldots, x_r$ or its reverse, depending on which direction we need to go. This is a contradiction to the condition. Therefore our assumption was wrong, and T is a tree.

---

16. Which complete bipartite graphs $K_{m,n}$, where m and n are positive integers, are trees?

16. If both m and n are at least 2, then clearly there is a simple circuit of length 4 in $K_{m,n}$. On the other hand, $K_{m,1}$ is clearly a tree (as is $K_{1,n}$). Thus we conclude that $K_{m,n}$ is a tree if and only if $m = 1$ or $n = 1$.

---

17. How many edges does a tree with 10,000 vertices have?

17. Since a tree with n vertices has $n - 1$ edges, the answer is 9999.

---

18. How many vertices does a full 5-ary tree with 100 internal vertices have?

18. By Theorem $4(ii)$, the answer is $mi + 1 = 5 \cdot 100 + 1 = 501$.

---

19. How many edges does a full binary tree with 1000 internal vertices have?

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.

---

20. How many leaves does a full 3-ary tree with 100 vertices have?

20. By Theorem $4(i)$, the answer is $[(m-1)n+1]/m = (2 \cdot 100 + 1)/3 = 67$.

---

21. Suppose 1000 people enter a chess tournament. Use a rooted tree model of the tournament to determine how many games must be played to determine a champion, if a player is eliminated after one loss and games are played until only one entrant has not lost. (Assume there are no ties.)

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.

---

22. A chain letter starts when a person sends a letter to five others. Each person who receives the letter either sends it to five other people who have never received it or does not send it to anyone. Suppose that 10,000 people send out the letter before the chain ends and that no one receives more than one letter. How many people receive the letter, and how many do not send it out?

22. The model here is a full 5-ary tree. We are told that there are 10,000 internal vertices (these represent the people who send out the letter). By Theorem $4(ii)$ we see that $n = mi + 1 = 5 \cdot 10000 + 1 = 50{,}001$. Everyone but the root receives the letter, so we conclude that 50,000 people receive the letter. There are $50001 - 10000 = 40{,}001$ leaves in the tree, so that is the number of people who receive the letter but do not send it out.

---

23. A chain letter starts with a person sending a letter out to 10 others. Each person is asked to send the letter out to 10 others, and each letter contains a list of the previous six people in the chain. Unless there are fewer than six names in the list, each person sends one dollar to the first person in this list, removes the name of this person from the list, moves up each of the other five names one position, and inserts his or her name at the end of this list. If no person breaks the chain and no one receives more than one letter, how much money will a person in the chain ultimately receive?

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.

***24.** Either draw a full m-ary tree with 76 leaves and height 3, where m is a positive integer, or show that no such tree exists.

24. Such a tree does exist. By Theorem 4(iii), we note that such a tree must have $i = 75/(m-1)$ internal vertices. This has to be a whole number, so $m-1$ must divide 75. This is possible, for example, if $m = 6$, so let us try it. A complete 6-ary tree (see preamble to Exercise 27) of height 2 would have 36 leaves. We therefore need to add 40 leaves. This can be accomplished by changing 8 vertices at level 2 to internal vertices; each such change adds 5 leaves to the tree (6 new leaves at level 3, less the one leaf at level 5 that has been changed to an internal vertex). We will not show a picture of this tree, but just summarize its appearance. The root has 6 children, each of which has 6 children, giving 36 vertices at level 2. Of these, 28 are leaves, and each of the remaining 8 vertices at level 2 has 6 children, living at level 3, for a total of 48 leaves at level 3. The total number of leaves is therefore $28 + 48 = 76$, as desired.

***25.** Either draw a full m-ary tree with 84 leaves and height 3, where m is a positive integer, or show that no such tree exists.

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.

***26.** A full m-ary tree T has 81 leaves and height 4.

a) Give the upper and lower bounds for m.

b) What is m if T is also balanced?

26. By Theorem 4(*iii*), we note that such a tree must have $i = 80/(m-1)$ internal vertices. This has to be a whole number, so $m-1$ must divide 80. By enumerating the divisors of 80, we see that m can equal 2, 3, 5, 6, 9, 11, 17, 21, 41, or 81. Some of these are incompatible with the height requirements, however.

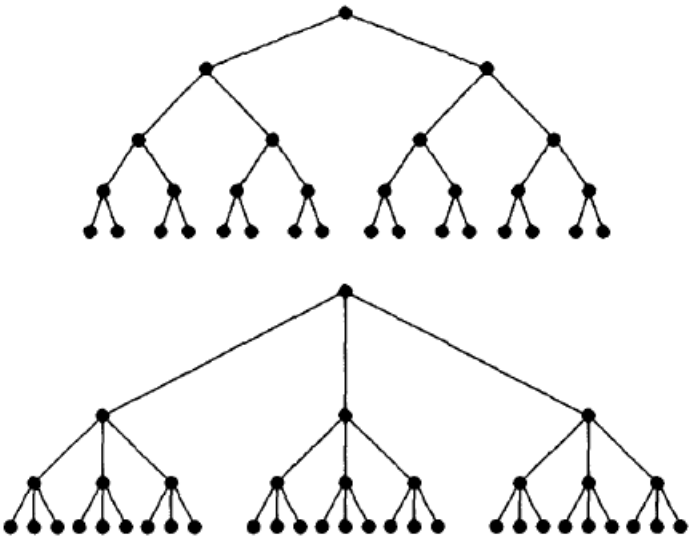
a) Since the height is 4, we cannot have $m = 2$, since that will give us at most $1+2+4+8+16 = 31$ vertices. Any of the larger values of m shown above, up to 21, allows us to form a tree with 81 leaves and height 4. In each case we could get m^4 leaves if we made all vertices at levels smaller than 4 internal; and we can get as few as $4(m-1)+1$ leaves by putting only one internal vertex at each such level. We can get 81 leaves in the former case by taking $m = 3$; on the other hand, if $m > 21$, then we would be forced to have more than 81 leaves. Therefore the bounds on m are $3 \le m \le 21$ (with m also restricted to being in the list above).

b) If T must be balanced, then the smallest possible number of leaves is obtained when level 3 has only one internal vertex and $m^3 - 1$ leaves, giving a total of $m^3 - 1 + m$ leaves in T. Again, the maximum number of leaves will be m^4. With these restriction, we see that $m = 5$ is already too big, since this would require at least $5^3 - 1 + 5 = 129$ leaves. Therefore the only possibility is $m = 3$.

A **complete m-ary tree** is a full m-ary tree in which every leaf is at the same level.

27. Construct a complete binary tree of height 4 and a complete 3-ary tree of height 3.

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.



---

28. How many vertices and how many leaves does a complete m-ary tree of height h have?

28. This tree has 1 vertex at level 0, m vertices at level 1, m^2 vertices at level 2, ..., m^h vertices at level h. Therefore it has

$$1 + m + m^2 + \cdots + m^h = \frac{m^{h+1} - 1}{m - 1}$$

vertices in all. The vertices at level h are the only leaves, so it has m^h leaves.

---

29. Prove

a) part (*ii*) of Theorem 4.

b) part (*iii*) of Theorem 4.

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 $n = i + l = [(l - 1)/(m - 1)] + l = (l - 1 + lm - l)/(m - 1) = (lm - 1)/(m - 1)$.

☞ **30.** Show that a full m-ary balanced tree of height h has more than m^{h-1} leaves.

30. (We assume $m \geq 2$.) First we delete all the vertices at level h; there is at least one such vertex, and they are all leaves. The result must be a complete m-ary tree of height $h - 1$. By the result of Exercise 28, this tree has m^{h-1} leaves. In the original tree, then, there are more than this many leaves, since every internal vertex at level $h - 1$ (which counts as a leaf in our reduced tree) spawns at least two leaves at level h.

---

31. How many edges are there in a forest of t trees containing a total of n vertices?

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.

---

32. Explain how a tree can be used to represent the table of contents of a book organized into chapters, where each chapter is organized into sections, and each section is organized into subsections.

32. The root of the tree represents the entire book. The vertices at level 1 represent the chapters—each chapter is a chapter of (read "child of") the book. The vertices at level 2 represent the sections (the parent of each such vertex is the chapter in which the section resides). Similarly the vertices at level 3 are the subsections.

---

34. What does each of these represent in an organizational tree?

a) the parent of a vertex

b) a child of a vertex

c) a sibling of a vertex

d) the ancestors of a vertex

e) the descendants of a vertex

f) the level of a vertex

g) the height of the tree

34. a) The parent of a vertex is that vertex's boss.

b) The child of a vertex is an immediate subordinate of that vertex (one he or she directly supervises).

c) The sibling of a vertex is a coworker with the same boss.

d) The ancestors of a vertex are that vertex's boss, his/her boss's boss, etc.

e) The descendants of a vertex are all the people that that vertex ultimately supervises (directly or indirectly).

f) The level of a vertex is the number of levels away from the top of the organization that vertex is.

g) The height of the tree is the depth of the structure.

---

35. Answer the same questions as those given in Exercise 34 for a rooted tree representing a computer file system.

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.

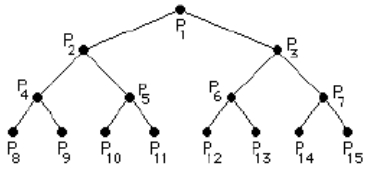
g) The height of the tree is the greatest depth (i.e., level) at which a file or directory is buried in the system.

---

36. **a)** Draw the complete binary tree with 15 vertices that represents a tree-connected network of 15 processors.

b) Show how 16 numbers can be added using the 15 processors in part (a) using four steps.

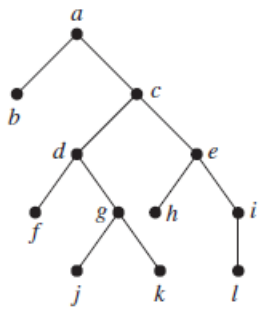
36. a) We simply add one more row to the tree in Figure 12, obtaining the following tree.



b) During the first step we use the bottom row of the network to add $x_1 + x_2$, $x_3 + x_4$, $x_5 + x_6$, ..., $x_{15} + x_{16}$. During the second step we use the next row up to add the results of the computations from the first step, namely $(x_1 + x_2) + (x_3 + x_4)$, $(x_5 + x_6) + (x_7 + x_8)$, ..., $(x_{13} + x_{14}) + (x_{15} + x_{16})$. The third step uses the sums obtained in the second, and the two processors in the second row of the tree perform $(x_1 + x_2 + x_3 + x_4) + (x_5 + x_6 + x_7 + x_8)$ and $(x_9 + x_{10} + x_{11} + x_{12}) + (x_{13} + x_{14} + x_{15} + x_{16})$. Finally, during the fourth step the root processor adds these two quantities to obtain the desired sum.

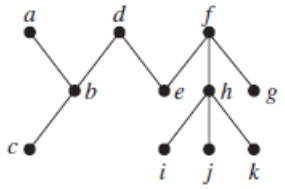
The **eccentricity** of a vertex in an unrooted tree is the length of the longest simple path beginning at this vertex. A vertex is called a **center** if no vertex in the tree has smaller eccentricity than this vertex. In Exercises 39–41 find every vertex that is a center in the given tree.

39.



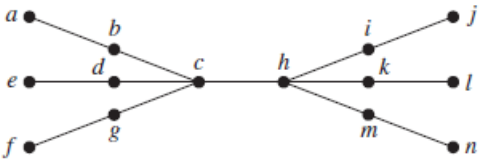
39. We need to compute the eccentricity of each vertex in order to find the center or centers. In practice, this does not involve much computation, since we can tell at a glance when the eccentricity is large. Intuitively, the center or centers are near the "middle" of the tree. The eccentricity of vertex 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.

---

40. 

40. The eccentricity of vertex e is 3, and it is the only vertex with eccentricity this small. Therefore e is the only center.

---

41. 

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.

---

42. Show that a center should be chosen as the root to produce a rooted tree of minimal height from an unrooted tree.

42. Since the height of a tree is the maximum distance from the root to another vertex, this is clear from the definition of center.

***43.** Show that a tree has either one center or two centers that are adjacent.

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.

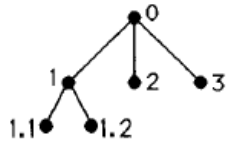
SECTION 11.3 Tree Traversal

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.

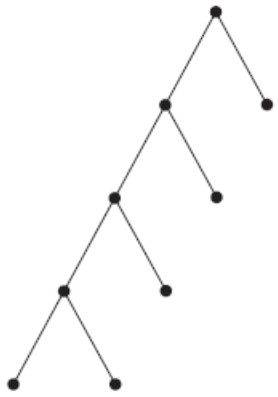
In Exercises 1–3 construct the universal address system for the given ordered rooted tree. Then use this to order its vertices using the lexicographic order of their labels.

1.

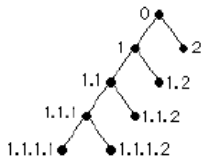
1. The root of the tree is labeled 0. The children of the root are labeled 1, 2, ..., from left to right. The children of a vertex labeled α are labeled $\alpha.1$, $\alpha.2$, ..., from left to right. For example, the two children of the vertex 1 here are 1.1 and 1.2. We completely label the tree in 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 $0 < 1 < 1.1 < 1.2 < 2 < 3$.



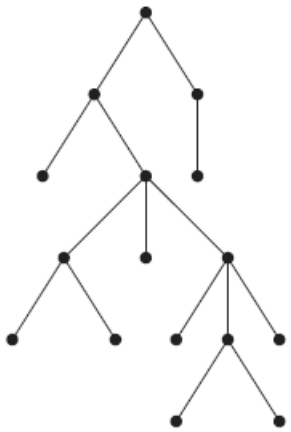
2.



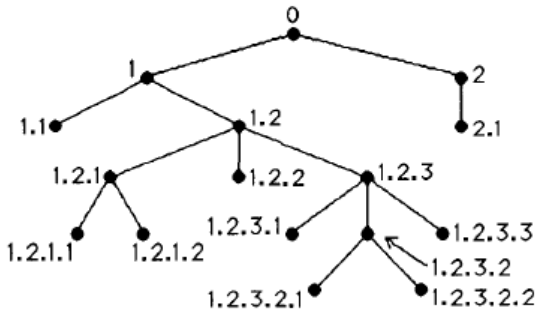
2. See the comments for the solution to Exercise 1. The order is $0 < 1 < 1.1 < 1.1.1 < 1.1.1.1 < 1.1.1.2 < 1.1.2 < 1.2 < 2$.



3.



3. See the comments for the solution to Exercise 1. The order is $0 < 1 < 1.1 < 1.2 < 1.2.1 < 1.2.1.1 < 1.2.1.2 < 1.2.2 < 1.2.3 < 1.2.3.1 < 1.2.3.2 < 1.2.3.2.1 < 1.2.3.2.2 < 1.2.3.3 < 2 < 2.1$.



---

4. Suppose that the address of the vertex v in the ordered rooted tree T is 3.4.5.2.4.

a) At what level is v?

b) What is the address of the parent of v?

c) What is the least number of siblings v can have?

d) What is the smallest possible number of vertices in T if v has this address?

e) Find the other addresses that must occur.

4. a) The vertex is at level 5; it is clear that an address (other than 0) of length l gives a vertex at level l.

b) We obtain the address of the parent by deleting the last number in the address of the vertex. Therefore the parent is 3.4.5.2.

c) Since v is the fourth child, it has at least three siblings.

d) We know that v's parent must have at least 1 sibling, its grandparent must have at least 4, its great-grandparent at least 3, and its great-great-grandparent at least 2. Adding to this count the fact that v has 5 ancestors and 3 siblings (and not forgetting to count v itself), we obtain a total of 19 vertices in the tree.

e) The other addresses are 0 together with all prefixes of v and the all the addresses that can be obtained from v or prefixes of v by making the last number smaller. Thus we have 0, 1, 2, 3, 3.1, 3.2, 3.3, 3.4, 3.4.1, 3.4.2, 3.4.3, 3.4.4, 3.4.5, 3.4.5.1, 3.4.5.2, 3.4.5.2.1, 3.4.5.2.2, and 3.4.5.2.3.

---

5. Suppose that the vertex with the largest address in an ordered rooted tree T has address 2.3.4.3.1. Is it possible to determine the number of vertices in T?

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.

---

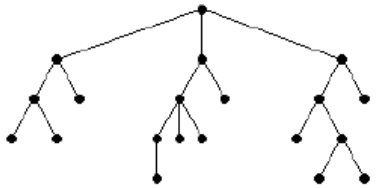
6. Can the leaves of an ordered rooted tree have the following list of universal addresses? If so, construct such an ordered rooted tree.

a) 1.1.1, 1.1.2, 1.2, 2.1.1.1, 2.1.2, 2.1.3, 2.2, 3.1.1, 3.1.2.1, 3.1.2.2, 3.2

b) 1.1, 1.2.1, 1.2.2, 1.2.3, 2.1, 2.2.1, 2.3.1, 2.3.2, 2.4.2.1, 2.4.2.2, 3.1, 3.2.1, 3.2.2

c) 1.1, 1.2.1, 1.2.2, 1.2.2.1, 1.3, 1.4, 2, 3.1, 3.2, 4.1.1.1

6. a) The following tree has these addresses for its leaves. We construct it by starting from the beginning of the list and drawing the parts of the tree that are made necessary by the given leaves. First of course there must be a root. Then since the first leaf is labeled 1.1.1, there must be a first child of the root, a first child of this child, and a first child of this latter child, which is then a leaf. Next there must be the second child of the root's first grandchild (1.1.2), and then a second child of the first child of the root (1.2). We continue in this manner until the entire tree is drawn.

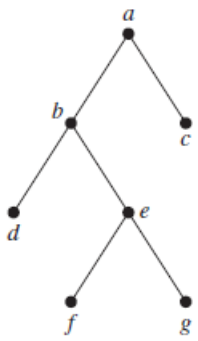


b) If there is such a tree, then the address 2.4.1 must occur since the address 2.4.2 does (the parent of 2.4.2.1). The vertex with that address must either be a leaf or have a descendant that is a leaf. The address of any such leaf must begin 2.4.1. Since no such address is in the list, we conclude that the answer to the question is no.

c) No such tree is possible, since the vertex with address 1.2.2 is not a leaf (it has a child 1.2.2.1 in the list).

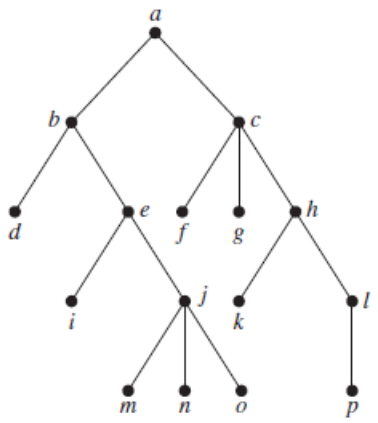
In Exercises 7–9 determine the order in which a preorder traversal visits the vertices of the given ordered rooted tree.

7.



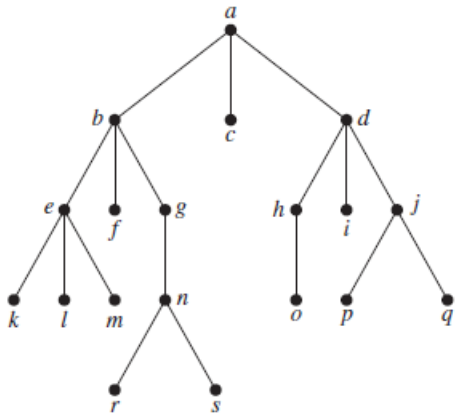
7. In preorder, the root comes first, then the left subtree in preorder, then the right subtree in preorder. Thus the preorder is a, followed by the vertices of the left subtree (the one rooted at b) in preorder, then c. Recursively, the preorder in the subtree rooted at b is b, followed by d, followed 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.

8.



8. 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 $a, b, d, e, i, j, m, n, o, c, f, g, h, k, l, p$.

9.



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 $a, b, e, k, l, m, f, g, n, r, s, c, d, h, o, i, j, p, q$.

---

10. In which order are the vertices of the ordered rooted tree in Exercise 7 visited using an inorder traversal?

10. The left subtree of the root comes first, namely the tree rooted at b. There again the left subtree comes first, so the list begins with d. After that comes b, the root of this subtree, and then the right subtree of b, namely (in order) f, e, and g. Then comes the root of the entire tree and finally its right child. Thus the answer is d, b, f, e, g, a, c.

11. In which order are the vertices of the ordered rooted tree in Exercise 8 visited using an inorder traversal?

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: $d, b, i, e, m, j, n, o, a, f, c, g, k, h, p, l$.

---

12. In which order are the vertices of the ordered rooted tree in Exercise 9 visited using an inorder traversal?

12. This is similar to Exercise 11. The answer is $k, e, l, m, b, f, r, n, s, g, a, c, o, h, d, i, p, j, q$.

---

13. In which order are the vertices of the ordered rooted tree in Exercise 7 visited using a postorder traversal?

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.

---

14. In which order are the vertices of the ordered rooted tree in Exercise 8 visited using a postorder traversal?

14. The procedure is the same as in Exercise 13, except that some vertices have more than two children here: before listing such a vertex, we list the vertices of its subtrees, in postorder, from left to right. The answer is $d, i, m, n, o, j, e, b, f, g, k, p, l, h, c, a$.

---

15. In which order are the vertices of the ordered rooted tree in Exercise 9 visited using a postorder traversal?

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 $k, l, m, e, f, r, s, n, g, b, c, o, h, i, p, q, j, d, a$.

---

16. **a)** Represent the expression $((x+2) \uparrow 3) * (y-(3+x)) - 5$ using a binary tree.
Write this expression in

b) prefix notation.
c) postfix notation.
d) infix notation.

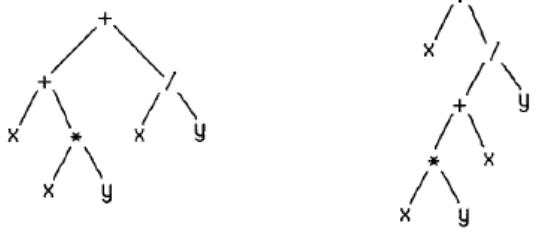
16. a) We build the tree from the top down while analyzing the expression by identifying the outermost operation at each stage. The outermost operation in this expression is the final subtraction. Therefore the tree has $-$ at its root, with the two operands as the subtrees at the root. The right operand is clearly 5, so the right child of the root is 5. The left operand is the result of a multiplication, so the left subtree has $*$ as its root. We continue recursively in this way until the entire tree is constructed.

---

17. **a)** Represent the expressions $(x + xy) + (x/y)$ and $x + ((xy + x)/y)$ using binary trees.
Write these expressions in

b) prefix notation.
c) postfix notation.
d) infix notation.

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. **a)** Represent $(A \cap B) - (A \cup (B - A))$ using an ordered rooted tree.

Write this expression in

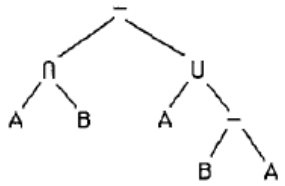
b) prefix notation.

c) postfix notation.

d) infix notation.

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: $((A\cap B)-(A\cup(B-A)))$.

---

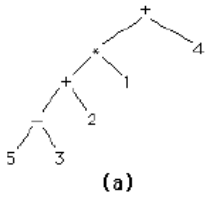
22. Draw the ordered rooted tree corresponding to each of these arithmetic expressions written in prefix notation. Then write each expression using infix notation.

a) $+*+-5\,3\,2\,1\,4$

b) $\uparrow+2\,3-5\,1$

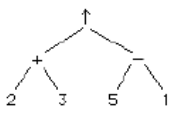
c) $*/9\,3+*2\,4-7\,6$

22. We work from the beginning of the expression. In part (a) the root of the tree is necessarily the first $+$. We then use up as much of the rest of the expression as needed to construct the left subtree of the root. The root of this left subtree is the $*$, and its left subtree is as much of the rest of the expression as is needed. We continue in this way, making our way to the subtree consisting of root $-$ and children 5 and 3. Then the 2 must be the right child of the second $+$, the 1 must be the right child of the $*$, and the 4 must be the right child of the root. The result is shown here.

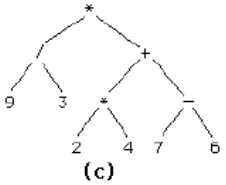


(a)

In infix form we have $((((5-3)+2)*1)+4)$. The other two trees are constructed in a similar manner.



(b)



(c)

The infix expressions are therefore $((2+3)\uparrow(5-1))$ and $((9/3)*((2*4)+(7-6)))$, respectively.

---

23. What is the value of each of these prefix expressions?

a) $- * 2 / 8 4 3$
b) $\uparrow - * 3 3 * 4 2 5$
c) $+ - \uparrow 3 2 \uparrow 2 3 / 6 - 4 2$
d) $* + 3 + 3 \uparrow 3 + 3 3 3$

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)

$$* + 3 + 3 \uparrow 3 + 3 3 3$$
$$* + 3 + 3 \uparrow 3 6 3$$
$$* + 3 + 3 729 3$$
$$* + 3 732 3$$
$$* 735 3$$
$$2205$$

24. What is the value of each of these postfix expressions?

a) $5\,2\,1\,-\,-\,3\,1\,4\,+\,+\,*$

b) $9\,3\,/\,5\,+\,7\,2\,-\,*$

c) $3\,2\,*\,2\,\uparrow\,5\,3\,-\,8\,4\,/\,*\,-$

24. We exhibit the answers by showing with parentheses the operation that is applied next, working from left to right (it always involves the first occurrence of an operator symbol).

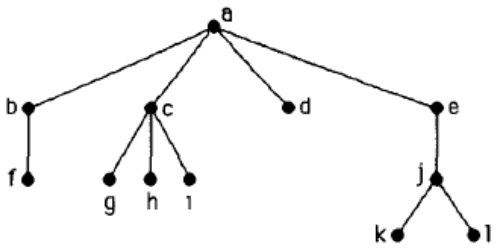
a) $5\,(2\,1\,-)\,-\,3\,1\,4\,+\,+\,*\,=\,(5\,1\,-)\,3\,1\,4\,+\,+\,*\,=\,4\,3\,(1\,4\,+)\,+\,*\,=\,4\,(3\,5\,+)\,*\,=\,(4\,8\,*)\,=\,32$

b) $(9\,3\,/)\,5\,+\,7\,2\,-\,*\,=\,(3\,5\,+)\,7\,2\,-\,*\,=\,8\,(7\,2\,-)\,*\,=\,(8\,5\,*)\,=\,40$

c) $(3\,2\,*)\,2\,\uparrow\,5\,3\,-\,8\,4\,/\,*\,-\;=\;(6\,2\,\uparrow)\,5\,3\,-\,8\,4\,/\,*\,-\;=\;36\,(5\,3\,-)\,8\,4\,/\,*\,-\;=\;36\;\;2\,(8\,4\,/)\,*\,-\;=\;36\,(2\,2\,*)\,-\;=\;(36\;\;4\,-)\,=\,32$

25. Construct the ordered rooted tree whose preorder traversal is *a, b, f, c, g, h, i, d, e, j, k, l*, where *a* has four children, *c* has three children, *j* has two children, *b* and *e* have one child each, and all other vertices are leaves.

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.



***26.** Show that an ordered rooted tree is uniquely determined when a list of vertices generated by a preorder traversal of the tree and the number of children of each vertex are specified.

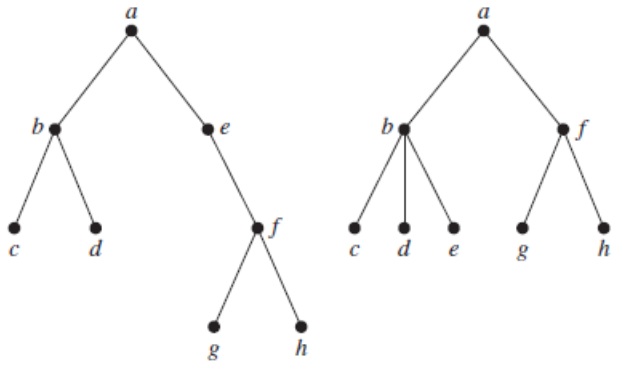
26. We prove this by induction on the length of the list. If the list has just one element, then the statement is trivially true. For the inductive step, consider the beginning of the list. There we find a sequence of vertices, starting with the root and ending with the first leaf (we can recognize the first leaf as the first vertex with no children), each vertex in the sequence being the first child of its predecessor in the list. Now remove this leaf, and decrease the child count of its parent by 1. The result is the preorder 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 first child of its parent (whom we know).

***27.** Show that an ordered rooted tree is uniquely determined when a list of vertices generated by a postorder traversal of the tree and the number of children of each vertex are specified.

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).

---

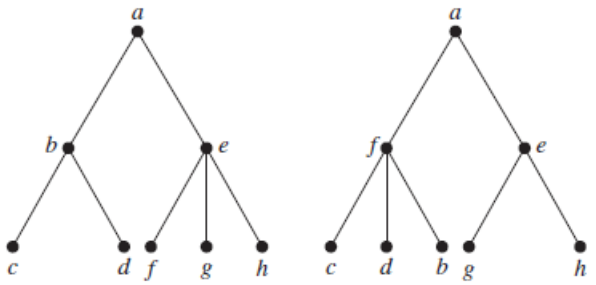
28. Show that preorder traversals of the two ordered rooted trees displayed below produce the same list of vertices. Note that this does not contradict the statement in Exercise 26, because the numbers of children of internal vertices in the two ordered rooted trees differ.



28. It is routine to see that the list is in alphabetical order in each case. In the first tree, vertex b has two children, whereas in the second, vertex b has three children, so the statement in Exercise 26 is not contradicted.

---

29. Show that postorder traversals of these two ordered rooted trees produce the same list of vertices. Note that this does not contradict the statement in Exercise 27, because the numbers of children of internal vertices in the two ordered rooted trees differ.



29. In each case the postorder is c, d, b, f, g, h, e, a.

Well-formed formulae in prefix notation over a set of symbols and a set of binary operators are defined recursively by these rules:

(i) if x is a symbol, then x is a well-formed formula in prefix notation;

(ii) if X and Y are well-formed formulae and $*$ is an operator, then $*XY$ is a well-formed formula.

30. Which of these are well-formed formulae over the symbols $\{x, y, z\}$ and the set of binary operators $\{\times, +, \circ\}$?

a) $\times + + x\, y\, x$

b) $\circ x\, y \times x\, z$

c) $\times \circ x\, z \times \times x\, y$

d) $\times + \circ x\, x \circ x\, x\, x$

30. a) This is not well-formed by the result in Exercise 31.

b) This is not well-formed by the result in Exercise 31.

c) This is not well-formed by the result in Exercise 31.

d) This is well-formed. Each of the two subexpressions $\circ xx$ is well-formed. Therefore the subexpression $+\circ xx\circ xx$ is well-formed; call it A. Thus the entire expression is $\times Ax$, so it is well-formed.

***31.** Show that any well-formed formula in prefix notation over a set of symbols and a set of binary operators contains exactly one more symbol than the number of operators.

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 that the statement is true for formulae of length less than $n > 1$, and let F be a formula of length n. Then F arises from part (ii) of the definition, so F consists of $*XY$, for some operator $*$ and some formulae X and Y. By the inductive hypothesis, the number of symbols in X exceeds the number of operators there by 1, and the same holds for Y. If we add and note that there is one more operator in F than in X and Y combined, then we see that the number of symbols in F exceeds the number of operators in F by 1, as well.

---

32. Give a definition of well-formed formulae in postfix notation over a set of symbols and a set of binary operators.

32. The definition is word-for-word the same as that given for prefix expressions, except that "postfix" is substituted for "prefix" throughout, and $*XY$ is replaced by $XY*$.

---

33. Give six examples of well-formed formulae with three or more operators in postfix notation over the set of symbols $\{x, y, z\}$ and the set of operators $\{+, \times, \circ\}$.

33. Any string of length n, using these six characters, is a well-formed formula as long as two conditions are met: if we read the string from left to right, the number of symbols is always at least 1 greater than the number of operators; and in all there is one more symbol than operator. We are asked to write down six such strings, with $n \geq 7$. One such set is $xxxx+++$, $xxxxx++++$, $xx+xx++$, $xxxx+xx+++$, $xxx+xx+++$, and $xx+xxx+++$.

---

34. Extend the definition of well-formed formulae in prefix notation to sets of symbols and operators where the operators may not be binary.

34. We replace the inductive step (ii) in the definition with the statement that if $X_1, X_2, \ldots, X_n$ are well-formed formulae and $*$ is an n-ary operator, then $*X_1X_2 \ldots X_n$ is a well-formed formula.