King Fahd University of Petroleum and Minerals
College of Computer Science and Engineering
Information and Computer Science Department

ICS 353: Design and Analysis of Algorithms

First semester 2020
Final Exam, Sunday, December 27, 2020.

Name:

ID#:

Instructions:

1. The exam consists of 8 pages, including this page, containing 6 questions. You have to answer all 6 questions.
2. The exam is closed book and closed notes. No calculators or any helping aides are allowed. Make sure you turn off your mobile phone and keep it in your pocket if you have one.
3. The maximum number of points for this exam is **100**.
4. You have exactly **120** minutes to finish the exam.
5. Make sure your answers are **readable**.
6. If there is no space on the front of the page, feel free to use the back of the page. Make sure you indicate this in order for me not to miss grading it.

Question Number	Maximum # of Points	Earned Points
1	10	
2	15	
3	15	
4	25	
5	20	
6	15	
Total	**100**	

***. Some Useful Formulas:**

$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$ $\sum_{i=1}^{n} i^3 = \left(\frac{n(n+1)}{2}\right)^2$

$\sum_{i=0}^{n} x^i = \frac{x^{n+1}-1}{x-1}$ $\sum_{i=1}^{n} \left(\frac{1}{2}\right)^i . i = 2 - \frac{n+2}{2^n}$ $2^{\lg n} = n$

,

$log_b a = \frac{log_c a}{log_c b}$ where $c, b \neq 1$ $\log a^b = b \log a$ $\log ab = \log a + \log b$

$\frac{d}{dx} \log_2 x = \frac{1}{\ln(2)x}$ $\frac{d}{dx} x^k = kx^{k-1}$ $\frac{d}{dx} k^x = \ln(k)k^x$

Master Theorem cases:

1- If $\exists \varepsilon > 0$, $g(n) = O(n^{\log_b a - \varepsilon})$, then $f(n) = \Theta(n^{\log_b a})$.

2- If $g(n) = \Theta(n^{\log_b a})$, then $f(n) = \Theta(n^{\log_b a} \log n)$.

3- If $\exists \varepsilon > 0$, $g(n) = \Omega(n^{\log_b a + \varepsilon}),$ and if $\exists c < 1$, $n_0 \in \aleph$, $ag(n/b) \leq cg(n)$ $\forall n > n_0$, then $f(n) = \Theta(g(n))$.

The characteristic equation of a linear homogeneous equations when k =2 : $x^2 - a_1 x - a_2 = 0$, and the closed form formulas: : $f(n) = c_1 r^n + c_2 n r^n$ if $r_1 = r_2$ or $f(n) = c_1 r_1^n + c_2 r_2^n$ if $r_1 \neq r_2$

Q1. (10 points; 5, 5 points) Consider the following divide and conquer algorithm:

```
Algorithm(A,low,high)
   if high - low = 1 then
   if A[low] < A[high] then return (A[low],A[high])
   else return (A[high],A[low])
   end if
   else
   mid =(low + high)/2
    (x1,y1) = Algorithm (A,low,mid)
    (x2,y2) = Algorithm (A,mid+1,high)
     x = min{x1,x2}
    y = max{y1,y2}
    return (x,y)
```

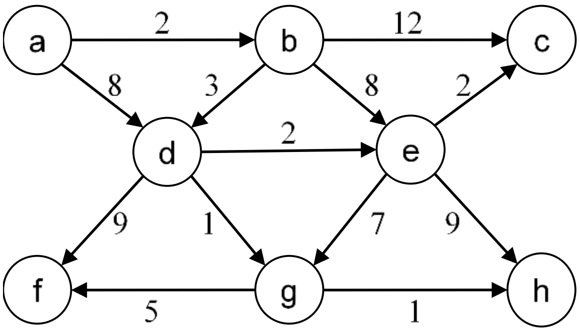
a. Formulate the cost of running time in a recurrence relation form. You may assume that the size of the array is a power of 2.

b. Express the running time of this algorithm in terms of O() notation.

Q2. (15 points) Run the dynamic programming solution of the knapsack problem on a knapsack of size 17 with the following items:

Item	Size	Value
1	3	5
2	4	7
3	5	8
4	8	11

Q3. **(15 points)** For the following graph, carry out Dijkstra's algorithm, starting from vertex a, to solve the single source shortest path problem. Show your work.

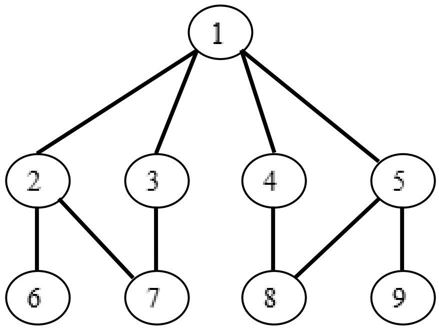


Q4. (25 points; 15, 5, 5 points):

a. Given an undirected graph G = (V, E), develop a greedy algorithm to find a vertex cover of minimum size.

b. What is the time complexity of your algorithm.

c. Apply your algorithm on the graph below and state whether it correctly finds it or not.



Q5. **(20 points)** Longest Path problem: Given a weighted graph $G = (V,E)$, two distinguished vertices $s, t \in V$ and a positive integer k, is there a simple path in G from s to t of length k or more?

Prove that the Longest Path problem is NP-Complete. (Hint: reduce the Hamiltonian Path problem to the Longest Path Problem)

Q6. (15 points) Apply the backtracking algorithm for the 3-coloring problem on the graph below. Make sure you show all intermediate steps.