C4-R3: ALGORITHM ANALYSIS AND DESIGN

NOTE:

1.	**Answer question 1 and any FOUR questions from 2 to 7.**
2.	**Parts of the same question should be answered together and in the same sequence.**

Time: 3 Hours **Total Marks: 100**

1.

a) Define the meaning of "Parallel Cost Optimal Algorithm."

b) Consider the evaluation of the product of n matrices

$M_1 * M_2 * \ldots\ldots * M_n$.

Assuming multiplication of p*q matrix and q*r matrix requires pqr operations, write an algorithm for ordering the above multiplication.

c) Define back-tracking. Describe an algorithm of some problem where backtracking is an intrinsic part of the algorithm.

d) Under what circumstances the condition "P=NP" is true?

e) Give an algorithm for finding an unknown number x in a sequence of supplied numbers. Find Θ, Ω and O of the algorithm.

f) What is the property that makes Kruskal's algorithm a unique greedy algorithm?

g) What are the differences between *heuristic* and approximation algorithms?

(7x4)

2.

a) What is an algorithm? What are the characteristics of an algorithm?

b) Show that Euclid's algorithm for computing GCD of a pair of positive integers has all the necessary properties of an algorithm.

c) How can you count number of ones in a binary string? Show that the counting algorithms for the above problem of a binary string of length n have time complexities varied from O(n) to O(1).

(4+6+8)

3.

a) State Cook's theorem. Describe time and space complexities.

b) What is polynomial time reducibility? Give example(s).

c) Differentiate among P, NP, NP-complete and NP-hard class of problems with suitable examples.

(4+4+10)

4.

a) Distinguish between divide-and-conquer and dynamic programming with suitable examples.

b) Describe a sorting algorithm that is designed based on the divide-and-conquer technique. Give a suitable example to illustrate the algorithm. Compute the worst-case computational complexity of the algorithm.

c) Show how a^m can be computed using dynamic programming technique, for any large positive integer value of m?

(4+8+6)

5.

a) What is the difference between the min-heap property and the binary search tree property? Can the min-heap property be used to print out the keys of a binary tree of n vertices in sorted order in O(n) time? Justify.

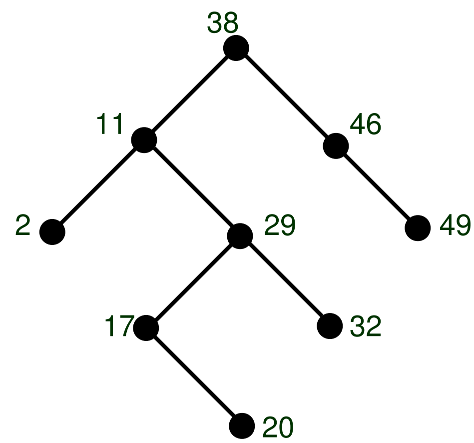
b) Devise a O(n+m) time algorithm for computing a component graph of a directed graph G=(V,E), where |V| = n and |E| = m. Make sure that your algorithm produces at most one edge between any pair of vertices in the component graph.

(10+8)

6.

a) Write an efficient algorithm to compute the third largest element of a list of n numbers, where $3 \leq n$. Deduce the computational time complexity of your algorithm and explain why are you calling it *efficient*.

b) Consider the following binary tree. Is it an AVL tree? If not, take the necessary steps to balance it.



Perform the following operations and at the end of each operation balance the tree in case it has lost the property of the AVL tree. Insert 33; Insert 51; Delete 20.

(9+9)

7.

a) What are *spanning tree* and *minimum spanning tree (MST)* of a graph?

b) *Traveling Salesman Problem (TSP)* finds a minimal cost tour in a connected weighted graph that touches every vertex of the graph exactly once and comes back to the initial vertex; show that the approximate solution of *TSP* is twice the *MST* of the graph.

(6+12)