C4-R4: ADVANCED ALGORITHMS

NOTE:

1. **Answer question 1 and any FOUR from questions 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) State the advantages and disadvantages of representing a graph as an adjacency-list.

b) What are the minimum and maximum numbers of elements in a heap of height h?

c) What is the difference between divide-and-conquer and dynamic programming strategy? What is the time and space complexity of matrix chain multiplication using dynamic programming strategy?

d) Solve the recurrence relation

$T(n) = O(1)$ if $n <= 1$

$T(n) = 2T(n/2) + O(1)$ otherwise

e) Define 0-1 knapsack problem and fractional knapsack problem. Give a sketch of how fractional knapsack can be solved using greedy strategy?

f) What is amortized analysis? What is its difference from average-case analysis?

g) Define the complexity class NP and NP-complete.

(7x4)

2.

a) Design a breadth-first search algorithm for an undirected graph G=(V, E). Illustrate it with an example. Derive the time complexity of the algorithm.

b) Suppose for a given graph G(V, E), a minimum spanning tree T=(V, E') is also given. Now one more node v is added to the graph. Node v is joined to two vertices (u, w) of the graph with a weighted edge for each vertex where $(u, w) \in E'$. Design a scheme for finding the minimum spanning tree of the new graph.

(9+9)

3.

a) Define the matrix-chain multiplication problem. How dynamic programming can be used to solve this problem.

b) State the various pivot choosing schemes in quick sort.

c) In binary search the sorted array is divided into almost two equal parts. What will be the time complexity of ternary search where the sorted array is divided into almost three equal parts?

(9+5+4)

4.

a) Define the minimum spanning tree problem.

b) State the steps of Kruskal's algorithm to solve the above problem. What kind of programming paradigm does it use?

c) Derive the time complexity of Kruskal's Algorithm.

d) What is the fundamental data structure used in Kruskal's Algorithm.

(3+7+4+4)

5.

a) Define the minimum vertex cover problem.
b) Why this is considered to be a hard problem.
c) What do you mean by a Polynomial Time Approximation Scheme for solving a NP-hard problem? What is approximation ratio in this context?
d) State the set cover problem.

(3+6+6+3)

6.

a) Write the pseudo code of how a set of n integers are sorted using mergesort.
b) Derive the message complexity of mergesort.
c) What is “in-place” sorting? Is mergesort an “in-place” sorting?
d) Why heap-sort is considered better than mergesort even though both have similar asymptotic time complexity?

(5+5+4+4)

7.

a) State the single source shortest path problem in undirected graph.
b) How this problem is solved using Dijkstra’s Algorithm?
c) What is the time complexity of Dijkstra’s Algorithm?
d) What is the traveling salesman problem? What kind of technique you need to solve this problem.

(4+6+4+4)