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 P, NP, NP-Hard SNP-Complete class of problems.

b) Use mathematical induction to show that when n is an exact power of 2, the solution of the recurrence

$$T(n) = \begin{cases} 2 & \text{if } n = 2 \\ 2T(n/2) + n & \text{if } n = 2^k \text{ for } k>1\} \end{cases}$$

is T(n)= n lg n.

c) Show that in any sub-tree of a max-heap, the root of the sub-tree contains the largest value occurring anywhere in the sub-tree.

d) What is the use of prefix function in Knuth-Morris-Pratt String Matching Algorithm?

e) A sequence of n operation is performed on a data structure. The *i* th operation cost *i* if *i* is an exact power of 3, and 1 otherwise. Use aggregate analysis to determine the amortized cost per operation.

f) Give an algorithm that determines whether or not a given undirected graph G = (V, E) contains a cycle. Your algorithm should run in *O* (V) time, independent of |E|.

g) Show that if a node in a binary search tree has two children, then its successor has no left child and its predecessor has no right child.

(7x4)

2.

a) Let R (i, j) be the number of times that table entry m[i, j] is referenced while computing other table entries in a call of MATRIX-CHAIN_ORDER. Show that the total number of references for the entire table is

$$\sum_{i=1}^{n} \sum_{j=i}^{n} R(i,J) = (n^3-n)/3 \ .$$

b) What is the difference between the binary search tree property and the min-heap property? Can the min-heap property be used to print out the keys of an n-node tree in sorted order in *O* (n) time? Explain why?

(10+8)

3.

a) How can we find out i^{th} smallest number from an array of *n* numbers in expected linear time?

b) Assuming 3-CNF satisfiability problem to be NP-Complete, prove clique problem is also NP-Complete.

(9+9)

4.

a) Show that a depth-first search of an undirected graph G can be used to identify the connected component of G and that the depth-first forest contains as many trees as G has connected components. More precisely, show how to modify depth-first search so that each vertex *v* is assigned an integer label *cc[v]* between 1 and *k*, where *k* is the number of connected components of G, such that *cc[u] = cc[v]* if and only *u* and *v* are in the same connected component.

b) Give an *O*(V+E) time algorithm to compute the component graph of a directed graph G=(V, E). Make sure that there is at most one edge between two vertices in the component graph your algorithm produces.

(10+8)

5.

a) Discuss Dijkastra's algorithm to solve the single source shortest paths on a weighted, directed graph with a example.

b) We are given a directed graph G = (V,E) on which each edge $(u,v) \in E$ has an associated value *r(u,v),* which is a real number in the range $0 \leq r(u,v) \leq 1$ that represents the reliability of a communication channel from vertex *u* to vertex *v.* We interpret *r(u,v)* as the probability that the channel from u to v will not fail, and we assume that these probabilities are independent. Give an efficient algorithm to find the most reliable path between two given vertices.

(8+10)

6.

a) Explain Ford-Fulkerson method to solve maximum flow problem. What is residual capacity?

b) A professor has two children who, unfortunately, dislike each other. The problem is to serve that not only do they refuse to walk to school together, but in fact each one refuses to walk on any block that the other child has stepped on that day. The children have no problem with their paths crossing at a corner. Fortunately, both the professor's house and the school are on corners, but beyond that he is not sure if it is going to be possible to send both of his children to the same school. The Professor has a map of his town. Show how to formulate the problem of determining if both his children can go to the same school as a maximum-flow problem.

(12+6)

7.

a) What is a convex hull? Discuss some methods that compute convex hulls in *O(n lg n)*time.

b) Briefly describe Graham's scan algorithm to solve the convex-hull problem.

([4+6]+8)