Ex/CSE/T/325A/2019

B.E. COMPUTER SCIENCE AND ENGINEERING THIRD YEAR SECOND SEMESTER EXAMINATION 2019

*DESIGN AND ANALYSIS OF ALGORITHMS*

Time: Three hours Full Marks: 100

Answer **FIVE** questions by choosing one question from each group

## GROUP A

1. a) Let $\alpha$ and $\beta$ be real numbers such that $0 < \alpha < \beta$. Show that $n^{\alpha}$ is in $O(n^{\beta})$, but $n^{\beta}$ is not in $O(n^{\alpha})$

   b) State Master theorem. For what type of algorithm, *Master theorem* is applicable. Give a detailed proof for *Master theorem*. 5 + 15 =20 marks

OR

2. a) Consider following function written in C style and compute the time complexity.

```
void fun1(int n)
{
 int i;
 for (i = 1;  i < n*n*n;  i = i*2)
 {
  // Do some O(1) works
 }
}
```

   b) Show that ${}^{n}C_{k} = O(n^{k})$. Use *Limit method* to prove it.

   c) Compute the value of the following recurrence equation using *recursion tree.*

   $T(n) \leq 3T(n/2) + n$ when $n>2$, and $T(2)\leq 1$ 5 + 5 +10 = 20 marks

## GROUP B

3. a) Differentiate (with a suitable example) between the following algorithm design strategies: "*Divide and Conquer*" and "*Dynamic Programming*".

   b) Recall the elementary –school level algorithm for multiplying integers, which computes a partial product by multiplying each digit of a number *x* separately by another number *y*, and then add up all partial products. Explain the time complexity of the elementary–school level algorithm. Write an efficient algorithm that uses *Divide and Conquer* Approach to this problem. Compare its time complexity with that of the elementary–school level algorithm. 5 +15 = 20 marks

OR

4. a) Explain the basic characteristics of greedy algorithm. Discuss with a suitable example.

   b) Consider the following coin changing problem and suggest a greedy algorithm for the problem: Given currency denominations: $C_1$, $C_2$, ---$C_n$, an amount X is to be paid to the customer using the fewest number of coins. Write a greedy algorithm for finding optimal set of coins which combine the amount X. Give a suitable example to show how the algorithm works. Does greedy algorithm always give optimal solution to coin changing problem?—explain. 5 + 15 =20 marks

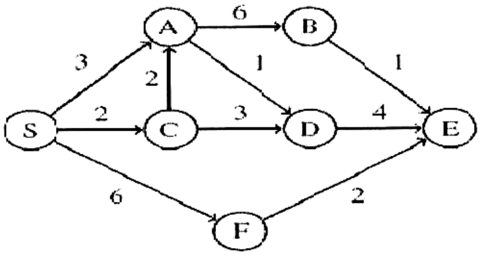
[ Turn over

## GROUP C

5. Differentiate between the time complexities of the linear queue based implementation and the priority queue based implementation of Dijkstra's shortest path algorithm.

   Apply Dijkstra's shortest path algorithm on the following graph and find the shortest paths of each vertex of the graph from the source vertex S. show also the order in which vertices get removed from the linear queue? What is the resulting shortest-path tree?

   5 + 15 = 20 marks



OR

6. A) Differentiate between the time complexities of linear queue based implementation and priority queue based implementation of Prim's algorithm for finding minimum spanning tree in a weighted graph.

   b) Write a randomized version of Quicksort algorithm and compute its expected running time.

   5+ 15 = 20 marks

## GROUP D

7. a) Prove correctness of the bubble sort algorithm.
   b) Formulate 0-1 Knapsack problem without repetition for solving it with Dynamic programming. (Show up to formulation only).
   d) What is the time complexity of dynamic programming based algorithm for 0-1 Knapsack problem. Is it linear in input size?—explain. 10 + 5+ 5=20 marks

OR

8. a) Consider the items, their values and weights as shown in the following table and find the solution to the 0-1 KNAPSACK problem with dynamic programming when knapsack capacity, W=8

| Item i | Value vi | Weight wi |
|---|---|---|
| 1 | 15 | 1 |
| 2 | 10 | 5 |
| 3 | 9 | 3 |
| 4 | 5 | 4 |

   b) Prove correctness for the Insertion sort algorithm. 10 + 10 =20 marks

## GROUP E

9. a) Consider that a minimum cost communication network will be set up among the major cities of a country in such a way that there will a communication path between any two cites. You may consider each city as a vertex and each communication link as the edge between two vertices. The weight of an edge is the cost of constructing the link represented by the edge. Every connected sub -graph that includes all the vertices represents a feasible solution. Under the assumption that all weights are non-negative, suggest an efficient algorithm for finding the optimal solution with minimum-cost. Formulate the problem as the graph problem first and discuss which and how the efficient graph algorithms can be used to solve this problem. How the running time of your algorithm differs from the brute force algorithm?

b) Is 2-SAT problem a NP-complete problem?-explain your answer with justifications.

c) Can a problem be in NP-hard, but not in NP? –Explain . 10 + 5 + 5 = 20 marks

OR

10. a) Discuss with a suitable example an efficient algorithm for string alignment problem. Explain how this algorithm is useful in designing a spell checker (Only write the outline of the spell checker algorithm).

b) Prove the following statement
*Suppose X is an NP-complete problem, then X is solvable in polynomial time if and only if $P = NP$*

c) Is Halting problem an NP-complete problem? -briefly explain.

10 + 5 + 5= 20 marks