Roll No. .....

**REF NO. 051273**

# M. Sc. Semester II Examination
# 2021-22

## COMPUTATIONAL SCIENCE AND APPLICATIONS

Paper No. CSA-202 : Design and Analysis of Algorithms

*Time* : 3 hours *Full Marks* : 70

**(Write your Roll No. at the top immediately on the receipt of this question paper)**

*The figures in the right-hand margin indicate marks*

Answer any **five** questions

**1.** *(a)* Briefly discuss the various asymptotic notations used for algorithmic analysis with suitable examples. 7

*(b)* Define Job Sequencing with Deadlines Problem. Briefly discuss the greedy approach to find the optimal solution for the job sequencing with deadlines problem. Consider the jobs, their deadlines and associated profits as

shown in the table :

| *Jobs* | J1 | J2 | J3 | J4 | J5 | J6 | J7 |
|---|---|---|---|---|---|---|---|
| *Deadlines* | 5 | 6 | 3 | 2 | 4 | 2 | 7 |
| *Profits* | 180 | 160 | 170 | 280 | 100 | 80 | 250 |

Answer the following questions :

*(i)* Write the optimal schedule.

*(ii)* Are all the jobs completed within the optimal schedule?

*(iii)* What is the maximum earned profit? 7

**2.** *(a)* Define recurrence relation and explain Master Theorem for solving recurrence relation. Solve the following recurrence relations by using the Master Theorem : 7

*(i)* $T(n) = 4T\left(\frac{n}{4}\right) + n \log n$

*(ii)* $T(n) = 9T\left(\frac{n}{3}\right) + \Theta(n)$

*(b)* Distinguish between internal and external sorting techniques by taking suitable examples. Write the recurrence relation of Merge Sort and

solve it by using Recursion-Tree Method. Sort the following list of elements by using Merge Sort techniques (showing each step) : 7

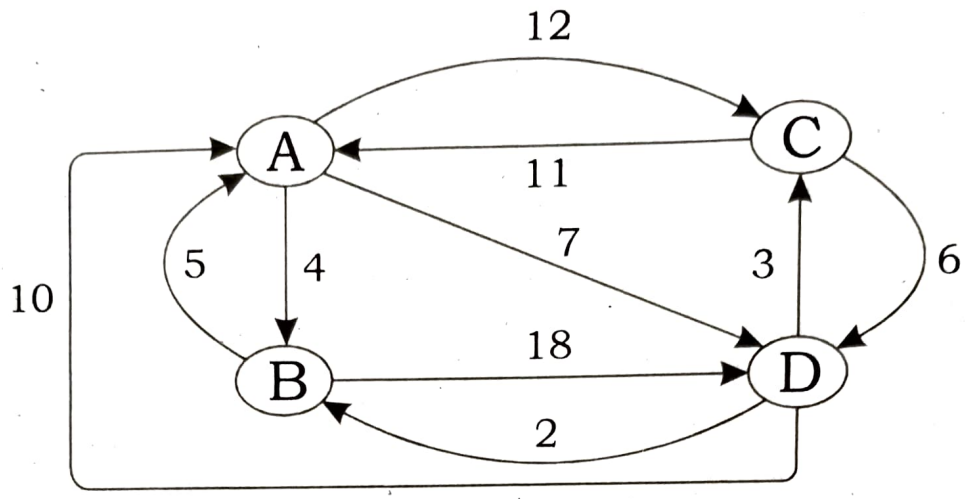
5, 32, 57, 4, 15, 77, 37, 46, 75, 99, 7, 25

**3.** Explain the working of 0/1 knapsack problem using dynamic programming approach. Is dynamic programming approach provides optimal solution for 0/1 knapsack problem? For the given set of items and knapsack capacity = 5 kg, find the optimal solution for the 0/1 knapsack problem making use of dynamic programming approach : 14

| *Item* | *Weight* | *Value* |
|---|---|---|
| 11 | 2 | 3 |
| 12 | 3 | 4 |
| 13 | 4 | 5 |
| 14 | 5 | 6 |

**4.** Explain traveling salesman problem. In the following directed weighted graph, find the optimal solution to the traveling salesman problem using Least Cost (LC)-Branch and Bound algorithm and construct a State

Space Tree for the solution : 14

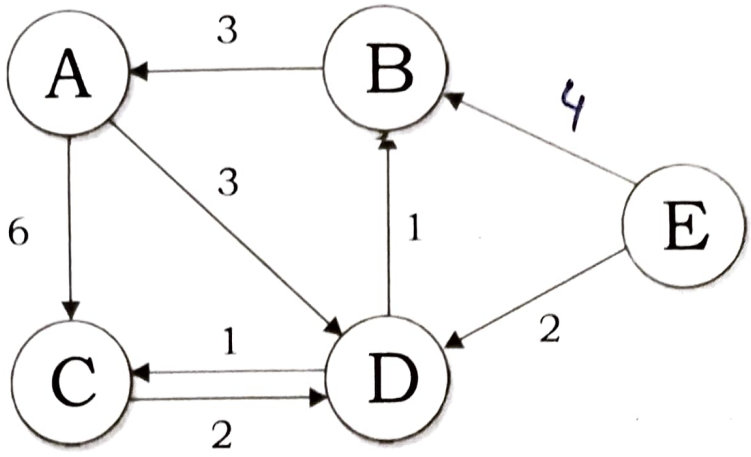



**5.** Explain Matrix Chain Multiplication Problem and find its time complexity. Find the optimal matrix chain parenthesization of the following given matrices by using dynamic programming approach and obtain the table M & S computed by the algorithm for the following matrices dimension : 14

$A1 : 10 \times 15$ $A2 : 15 \times 20$ $A3 : 20 \times 25$
$A4 : 25 \times 30$ $A5 : 30 \times 5$

**6.** *(a)* Why does Dijkstra's algorithm fail to find the Single Source Shortest Path problem for general weighted graphs? Discuss it with appropriate example. Explain the working of Bellman-Ford algorithm. Find the single source (Source $= B$) shortest path in the

weighted directed graph given below : 9




*(b)* Distinguish between *P*, *NP*, *NP*-complete and *NP*-hard problem by taking appropriate examples. 5

**7.** *(a)* How is the backtracking approach different from the Branch and Bound approach? Explain the *N*-Queen problem and its bounding function. Find all solutions of 4-Queen problem using backtracking approach and build its state space tree. 8

*(b)* Define algorithm and discuss its characteristics. Suppose there are two algorithms that are executed in a supercomputer (speed $= 10^9$ inst/sec) and personal computer (speed $= 10^7$ inst/sec) with time complexity $O(n^2)$ and $O(n \log_2 n)$ respectively. Find the time required to execute $10^6$ instruction by supercomputer and personal computer respectively and also justify why one algorithm is better than other. 6

**8.** Write short notes on *any two* of the following : 2×7

*(a)* Lower Bound Theory

*(b)* Strassen's Matrix Multiplication

*(c)* Floyd Warshall Algorithm

*(d)* Minimum Spanning Tree Algorithms.

★★★