(T) Roll No. .....

PAPER ID—11159

B.Tech. EXAMINATION, 2024

(Fifth Semester)

COMPUTER SCIENCE AND ENGINEERING

Design and Analysis of Algorithms

Time : 3 *Hours* *Maximum Marks :* 70

Before answering the question-paper candidates should ensure that they have been supplied to correct and complete question-paper. No complaint, in this regard, will be entertained after the examination.

Note : Attempt *Five* questions in all, selecting *one* question from each Unit. Q. No. 1 is compulsory. All questions carry equal marks.

1. Short answer type questions : 7×2=14

(a) Differentiate between Big O, Omega, and Theta notations.

(b) How many queues are required to implement a stack ? Justify.

(c) Can all Divide and Conquer algorithms be solved using Dynamic Programming ? Explain.

(d) What happens if two jobs have the same deadline in Job Sequencing ?

(e) Differentiate between top-down and bottom-up approaches in Dynamic Programming.

(f) Define live node, e-node, and dead node in Branch and Bound.

(g) Can NP-complete problems be solved in polynomial time ? Why or why not ?

Unit I

2. (a) What are time and space complexities ? How are they calculated for an algorithm ? Illustrate with examples. 7

   (b) Explain the Strassen's Matrix Multiplication algorithm. What is its time complexity, and how does it improve the standard matrix multiplication method ? 7

3. (a) What is the Union-Find (Disjoint Set Union) algorithm ? Explain its operations with examples. 7

   (b) What is the best, average, and worst-case time complexities of quicksort ? Explain the working of quicksort and how it achieves divide and conquer ? 7

Unit I

2. (a) What are time and space complexities ? How are they calculated for an algorithm ? Illustrate with examples. 7

   (b) Explain the Strassen's Matrix Multiplication algorithm. What is its time complexity, and how does it improve the standard matrix multiplication method ? 7

3. (a) What is the Union-Find (Disjoint Set Union) algorithm ? Explain its operations with examples. 7

   (b) What is the best, average, and worst-case time complexities of quicksort ? Explain the working of quicksort and how it achieves divide and conquer ? 7

Unit II

4. (a) Explain, how the Greedy Method is used to solve the Minimum Cost Spanning Tree problem. Compare Prim's and Kruskal's algorithms. 7

   (b) Define Knapsack problem. Solve the following 0/1 Knapsack problem using dynamic programming Let $n = 3$, $p = \{1, 2, 5\}$, $w = (2, 3, 4)$ and $m = 6$. 7

5. (a) Using Dynamic Programming, solve a Traveling Salesperson problem for the given graph : A $\rightarrow$ B (10), B $\rightarrow$ C (15), C $\rightarrow$ D (20), D $\rightarrow$ A (25), and other cyclic paths. Show all intermediate steps. 7

(b) Describe Dijkstra's algorithm for single-source shortest paths. How does it use the Greedy Method ? Provide an example. 7

Unit III

6. (a) Explain the general method of Backtracking. Apply it to solve the 8-Queens problem, demonstrating all intermediate steps for placing queens on a 4 × 4 chessboard. 7

(b) Discuss the limitations of Backtracking and Branch and Bound for large-scale optimization problems. Discuss the enhancements or alternative approaches to improve efficiency. 7

(b) Describe Dijkstra's algorithm for single-source shortest paths. How does it use the Greedy Method ? Provide an example. 7

Unit III

6. (a) Explain the general method of Backtracking. Apply it to solve the 8-Queens problem, demonstrating all intermediate steps for placing queens on a 4 × 4 chessboard. 7

(b) Discuss the limitations of Backtracking and Branch and Bound for large-scale optimization problems. Discuss the enhancements or alternative approaches to improve efficiency. 7

7. (a) Write an algorithm of sum of subsets problem. Solve the following problem and draw portion of state space tree M = 35, W = (5, 7, 10, 12, 15, 18, 20) 7

(b) Discuss efficiency considerations in Branch and Bound algorithms. How can the bounding function and data structures impact performance ? 7

Unit IV

8. (a) Define NP-Hard and NP-Complete problems. Explain their significance in computational complexity theory. Use examples to illustrate the difference between the two classes. 7

(b) Explain Cook's Theorem and its importance. 7

9. (a) What do you mean by NP-Hard graph problems ? Provide an example of a NP-Hard graph problem. 7

(b) What do you mean by simplified NP-Hard problems ? Provide an example of a simplified NP-hard problem and explain why it is still computationally challenging ? 7