JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TEST -3 EXAMINATION- 2025

B. Tech- IV Semester (CSE/IT/BT/BI)

COURSE CODE (CREDITS): 18B11CI412(3)

MAX. MARKS: 35

COURSE NAME: Design and Analysis of Algorithms

COURSE INSTRUCTORS: Dr. Aman Sharma, Dr. Arvind Kumar, Mr. Ravi Sharma, Mr. Saurav Singh

MAX. TIME: 2 Hours

Note: (a) All questions are compulsory.

(b) The candidate is allowed to make Suitable numeric assumptions wherever required for solving problems

Q.No	Question	CO	Marks
Q1	**1.1.** a) Explain how the Floyd-Warshall algorithm detects the presence of a negative weight cycle in a graph. **(2 marks)** b) Consider the following directed graph with 4 vertices. Use the Floyd-Warshall algorithm to compute the final distance matrix. Also, state whether the graph contains a negative weight cycle. **(3 marks)** 0 1 2 3 0 [0 1 ∞ ∞] 1 [∞ 0 -1 ∞] 2 [∞ ∞ 0 -1] 3 [-1 ∞ ∞ 0] **1.2** Given the recurrence relation: **$T(n) = 2T(n/2) + n \log n$** a) Use the **Recursion Tree Method** to guess the time complexity of the recurrence. Show the pattern clearly. **(2 marks)** b) State whether **Master's Theorem** can be applied to this recurrence. If yes, solve it using the theorem; if not, explain why. **(2 marks)**	4	9
Q2	**2.1** You are given the following weighted undirected graph: Vertices: {A, B, C, D, E} Edges with weights: A - B (1), A - C (3), B - C (1), B - D (4), C - D (1), D - E (2), C - E (5) a) Using Kruskal's algorithm, construct the Minimum Spanning Tree (MST). Show the order in which edges are added and justify cycle checks using the Union-Find approach. **(3 marks)** b) Using Prim's algorithm, construct the MST starting from vertex A. Show the step-by-step process of selecting the next minimum edge and the vertices included at each step. **(3 marks)** **2.2**	3	9

	a) Write the average case time complexity of the following sorting algorithms: i) Bubble Sort **(1 mark)** ii) Insertion Sort iii) Selection Sort iv) Heap Sort b) Briefly explain why Heap Sort is more efficient than Bubble Sort for large datasets. **(1 mark)** c)Which of the above algorithms is best suited for nearly sorted data, and why? **(1 mark)**		
	3.1 a) Define the following complexity classes with clear distinctions: **(3 marks)** P, NP, NP-complete, NP-hard b) Suppose a new algorithm is discovered that solves an NP-complete problem in polynomial time. i. What are the implications of this discovery on the P vs NP question? (3 marks) ii. What effect does it have on all other NP-complete problems? iii. Can NP-hard problems now be solved in polynomial time? Justify your answer. **3.2** a) Identify which algorithm design paradigm (Divide and Conquer, Greedy, or Dynamic Programming) is best suited for solving the following problems. Justify your answer briefly. **(1 mark)** i) Merge Sort ii) Fractional Knapsack Problem b) Briefly explain how the Divide and Conquer approach is applied in solving the Maximum Subarray Problem (Kadane's algorithm). **(1 mark)**	2	8
	4.1 You are given the following items, each with a weight and a value, and a knapsack with a capacity of 5 units: **Item Weight Value** 1 1 8 2 3 6 3 4 12 a) Construct the dynamic programming table used to solve the 0/1 Knapsack problem for the given data. **(3 marks)** b) From the table, determine the maximum value that can be obtained and list the items included in the optimal solution. **(2 marks)** **4.2** a) List any three essential characteristics of a good algorithm. **(1 mark)** b) Define the following asymptotic notations and explain what aspect of algorithm performance they represent: i) Big-O (O) ii) Big-Ω (Omega) iii) Big-Θ (Theta) **(2 mark)** c) Describe the difference between best-case, average-case, and worst-case time complexity with an example. **(1 mark)**	3	9