JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TEST -2 EXAMINATION- 2025

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

COURSE CODE (CREDITS): 18B11CI412(3)

MAX. MARKS: 25

COURSE NAME: Design and Analysis of Algorithms

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

MAX. TIME: 1 Hour 30 Min

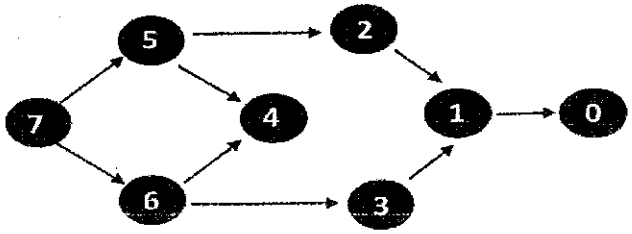
Note: *(a) All questions are compulsory.*

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

Q.No	Question	CO	Marks
Q1	a) Given five matrices **A_1 (10×20), A_2 (20×30), A_3 (30×40), A_4 (40×50), and A_5 (50×60)**, compute the **minimum number of scalar multiplications** required to multiply them using the Matrix Chain Multiplication (MCM) algorithm. Provide a **step-by-step explanation**, including: 1. **Defining the cost function** for computing optimal multiplication order. 2. **Constructing the DP table** and determining the minimum cost. 3. **Identifying the optimal parenthesization** of the matrix sequence. b) Explain the difference between **Memoization** and **Tabulation** in **Dynamic Programming**.	3	**[5+ 2]**

Q2	a) Calculate the Time and Space Complexity of below mentioned codes
	
	`void fun(int n) { for (int i = 1; i <= n; i *= 2) { for (int j = 0; j < i; j++) { cout << i << " " << j << endl; } } }`
	
	`def mystery(n): if n <= 1: return 1 return mystery(n // 2) + mystery(n // 3) + n`
	
	`void compute(int n) { if (n <= 1) return;`
	
	`int divide(int n) { if (n <= 1) return 1;`
		1	**[4+ 3]**

		for (int i = 0; i < n; i++) {
		System.out.println(i);
		}
		compute(n / 2);
		compute(n / 3);
		}
			return divide(n/4) + divide(n/4) + divide(n/4) + divide(n/4) + n;
			}		

		b) Solve the below mentioned recurrence relation with recursive tree method. **$T(n) = T(n/4) + T(n/2) + cn^2$**		
Q3	a)	Given **N jobs**, where each job ii has **Start time** S_i **Finish time** F_i and **Profit** P_i. You must select **non-overlapping jobs** to maximize total profit. 1. Propose an efficient algorithm to find the maximum profit. 2. Analyse its time complexity using **DP with Binary Search**. 3. Find the **maximum profit** for: Jobs=[(1,3,50),(2,5,20),(6,19,100),(7,8,200)]	4	**[3+ 3]**
	b)	Explain the key differences between Breadth-First Search (BFS) and Depth-First Search (DFS) with respect to their traversal order and memory usage. Provide a brief example of an application where one would be preferred over the other.		
Q4		Given a **directed acyclic graph (DAG)** with **N vertices** and **M edges**, some tasks must be completed in a specific order. 1. Explain the **Kahn's Algorithm** and **DFS-based approach** for topological sorting. 2. Analyze their **time and space complexity**. 3. Given the following graph, find a valid **topological order**: 	3	**[5]**

**********Best of Luck**********