JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TEST -2 EXAMINATION- 2025

M. Tech-II (CSE/IT)

COURSE CODE (CREDITS): 10M11CI211 (3)

MAX. MARKS: 25

COURSE NAME: Advanced Algorithms

COURSE INSTRUCTORS: Dr. Aman Sharma

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	Given two strings X and Y of lengths m and n, the Longest Common Subsequence (LCS) is the longest sequence that appears in both strings in the same order but not necessarily consecutively. a) Derive the recursive formula for LCS and explain its significance. b) Implement an efficient DP-based approach for finding LCS and analyze its time and space complexity. c) Compute the LCS length for the given strings: X="AGGTAB",Y="GXTXAYB"	3	[5]
Q2	A connected, undirected graph with N vertices and M edges is given. a) Explain how BFS and DFS traverse a graph differently. When does BFS perform better than DFS? b) Given the following adjacency list, perform both BFS and DFS starting from vertex 1 and write their respective order of traversal: Graph Adjacency List: 1→2,1→3,2→4,2→5,3→6,3→71 \to 2, 1 \to 3, 2 \to 4, 2 \to 5, 3 \to 6, 3 \to 7 c) If the graph were a weighted graph, why wouldn't BFS be ideal for finding the shortest path? Which algorithm would be preferable instead?	4	[5]

Q3	Analyze the time and space complexity of the following code:
	
	void compute(int n) {
	int count = 0;
	for (int i = 1; i <= n; i *= 2) { // Loop 1
	for (int j = 0; j < n; j++) { // Loop 2
	count++;
	}
		1	[5]

```
    }
    for (int k = 0; k < n; k++) {  // Loop 3
        count++;
    }
}
```

a) Determine the time complexity of the function step by step.
b) Find the space complexity of the function.
c) If n = 1024, approximately how many iterations does the function execute?
d) Can the time complexity be improved? If yes, suggest an optimization.

Q4	You are given N matrices, where the i^{th} matrix has dimensions A[i−1]×A[i]. The goal is to find the optimal order of multiplication to minimize the total number of scalar multiplications. a) Derive the recursive formula for the Matrix Chain Multiplication (MCM) problem. b) Explain how Dynamic Programming (DP) improves efficiency over recursion. c) Compute the minimum number of scalar multiplications required for multiplying the matrices with dimensions: A=[10,30,5,60] d) Analyze the time and space complexity of the DP approach. e) If four matrices are multiplied optimally, how many different ways exist to parenthesize them?	3	[5]
Q5.	a) Explain the Huffman Encoding algorithm and why it produces an optimal prefix code. b) Construct the Huffman Tree and determine the Huffman codes for the given character frequencies: **Character Frequency** A 5 B 9 C 12 D 13 E 16 F 45 c) Compute the total bits required to encode "ABCDEF". d) Analyze the time complexity of Huffman Encoding. e) Compare Huffman Encoding with fixed-length encoding in terms of efficiency.	3	[5]

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