

Quiz-1 Examination: Summer-2025

Course Code: CIS 212 (Batch: 20)

Course Title: Algorithm

Time: 30 Minutes

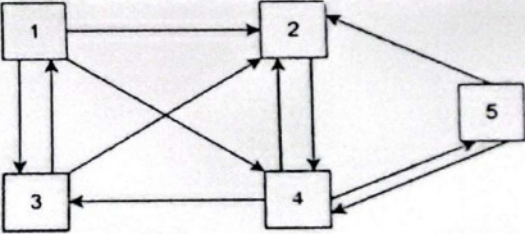
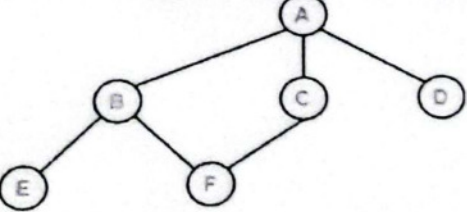
Total Marks: 15

1.	<p>What is complexity? Determine the time complexity of the following code using Big-O notation.</p> <p>A. <pre>for (int i = 0; i < m; i++) { for (int j = 0; j < n; j++) { for (int k = 0; k < 1000; k++) { cout << i * j * k << endl; } } }</pre></p> <p>B. <pre>for (int i = n; i > 0; i /= 2) { for (int j = 0; j < n; j++) { cout << i + j << endl; } }</pre></p>	5
2	Trace the steps to sort [38, 27, 43, 3, 9, 82, 10] using Merge Sort . Show the recursion tree for splitting and merging.	5
3	<p>Is Merge Sort an InPlace algorithm? Give a comparative analysis of Insertion, Merge & Quick Sort for:</p> <p>A. Best Case.</p> <p>B. Worst Case.</p>	5

Quiz-2 Examination: Summer-2025
Course Code: CIS 212 (Batch: 20)
Course Title: Algorithm

Time: 30 Minutes

Total Marks: 15

1.	<p>A binary tree T has 9 nodes. The inorder and preorder traversals of T yield the following sequences of nodes:</p> <p>Inorder: E A C K F H D B G</p> <p>Preorder: F A E K C D H G B</p> <ol style="list-style-type: none"> Draw the tree T. Find the order of tree T. Listed the leaf nodes of tree T. Listed the siblings of node K. Show the internal vertices of the tree. 	5
2	<p>Define DAG(Directed Acyclic Graph).</p> <p>Construct an adjacency matrix and adjacency list for the directed graph.</p> 	5
3	 <p>Apply Breadth First Search (BFS) and Depth First Search(DFS) algorithm and find the order of visiting nodes. Start Node= A.</p>	5

Quiz-3 Examination: Summer-2025

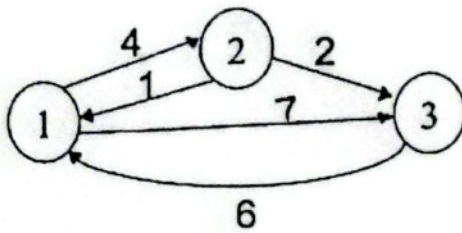
Course Code: CIS 132 (Batch: 20)

Course Title: Algorithm

Time: 30 Minutes

Total Marks: 15

1.



Compute the initial matrices $A^{(0)}$ and final matrices $A^{(k)}$ to determine the all pair shortest paths using **Floyd Warshall Algorithm**.

5

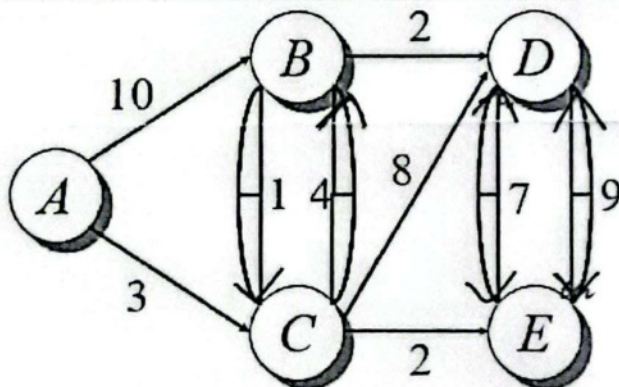
2

How does the **Bellman-Ford** algorithm detect negative-weight cycles? Explain the condition used for this check.

What do you mean by **Relaxation**? Write the pseudocode for **Relaxation**.

5

3



Apply **Dijkstra** algorithm on the given graph. Source node=A. Show the Queue states and Order of visiting nodes.

5