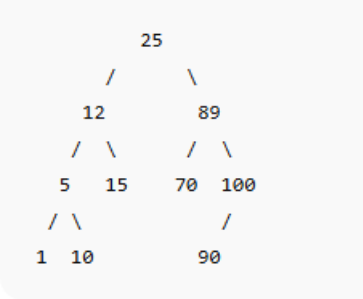
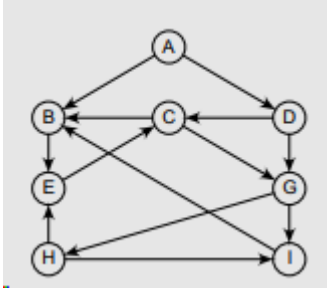
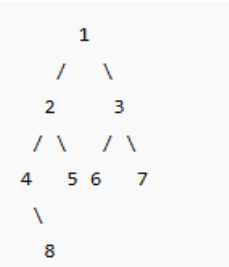
| (Affiliated to University of Mumbai) | ***End Semester Examination (R-24) SH 2025*** | | | |
|---|---|---|---|---|
| Branch: Computer Engineering/ AIDS/ AIML | Course: Data Structure | | | |
| Year/ Semester: SE / III | Course code: CEC302 | | | |
| Time: 03 hours | Marks: 80 | | | |
| **Note: 1. All questions are compulsory.**<br>**2. Figures to right indicate full marks.**<br>**3. Assume suitable data wherever necessary.** | | Marks | CO | BL |
| Q. 1 | Attempt any FOUR. (All questions carry equal marks) | **20** | | |
| A. | Differentiate between linear and non-linear data structures.<br>Definition of Data Structure – 01 Mark<br>Difference between Linear & Non Linear Data structure (4 Points) & Diagrammatic Representation – 04 Marks | 05 | CO1 | L4 |
| B. | Convert the expressions into its corresponding postfix expression using stack-(A + B) * (C + D)<br>All steps involve correct push and pop operations from the stack & conversion to a postfix expression. – 05 Marks<br>Construct a table. | 05 | CO2 | L3 |
| C. | Differentiate between a linked list and an array.<br>Five Comparison Points. – 05 Marks | 05 | CO3 | L4 |
| D. | Create a binary search tree by inserting the following elements in the given order: 25, 12, 89, 70, 100, 5, 15, 1, 10, 90.<br>  <br>Construct Root Node – 01 Mark<br>Construct Left Sub Tree – 02 Marks<br>Construct Right Sub Tree – 02 marks | 05 | CO4 | L3 |
| E. | Explain following terminologies of a graph:<br>a. Adjacent nodes<br>b. Path<br>c. Simple path<br>d. Cycle<br>e. Degree of a vertex<br>Each terminology carries 01 Mark.<br>Explain along with the graph. | 05 | CO5 | L2 |
| F. | Differentiate between linear and binary search.<br>Five Comparison Points with example – 05 Marks | 05 | 6 | 4 |
| | | | | |
| Q.2 | Attempt any FOUR. (All questions carry equal marks) | **40** | | |

| | | | | |
|---|---|---|---|---|
| A. | Write a C program to simulate the customer queue in a bank using a queue data structure. | 10 | CO2 | L3 |
| B. | Write a C program to store student records (rollno, name, marks) using a linked list and perform operations- 1. insert a new record at the end, and 2. calculate and display the average marks | 10 | CO3 | L3 |
| C. | Construct an **AVL Tree** by inserting the following sequence of keys into an initially empty tree: 10, 20, 30, 25, 28, 27, 5<br>i. Tree after each insertion – 05 Marks<br>ii. Clearly indicate where rotations are performed to maintain the balance factor- 03 Marks<br>iii. Specify the **type of rotation** used in each step (LL, RR, LR, or RL) - 02 Marks | 10 | CO4 | L3 |
| D. | Consider the graph given below. Apply depth-first and breadth-first traversal schemes for the following graph.<br>  <br><br>DFS Traversal Method using Stack Data Structure step-by-step procedure – 05 Marks<br>DFS traversal: A → B → E → C → G → I → H → D<br>BFS Traversal Method using Queue Data Structure step-by-step procedure – 05 Marks<br>BFS traversal: A → B → C → D → E → H → G → I | 10 | CO5 | L3 |
| E. | Consider a hash table with size = 10. Using quadratic probing, insert the keys 27, 72, 63, 42, 36, 18, 29, and 101 into the table. Take c1 = 1 and c2 = 3.<br>Calculate hash<br>Value – 08 Marks<br>Collisions – 02 Marks<br><br>**Index 0 1 2 3 4 5 6 7 8 9**<br>Key 36 101 72 63 42 27 18 29 | 10 | CO6 | L3 |
| F. | Construct a Huffman tree and determine the code for the following characters whose frequencies are given: A:20, B:10, C:10, D:30, E:30.<br>Construct the Huffman tree step by step – 05 Marks<br>Code for each symbol - 05 Marks | 10 | CO4 | L3 |

| | | | | |
|---|---|---|---|---|
| | <table><tr><th>Symbol</th><th>Freq</th><th>Code</th></tr><tr><td>A</td><td>20</td><td>00</td></tr><tr><td>B</td><td>10</td><td>010</td></tr><tr><td>C</td><td>10</td><td>011</td></tr><tr><td>D</td><td>30</td><td>10</td></tr><tr><td>E</td><td>30</td><td>11</td></tr></table><br>A : 00<br>B : 010<br>C : 011<br>D : 10<br>E : 11 | | | |
| | | | | |
| Q.3 | Attempt any FOUR | **20** | | |
| A. | Write an algorithm to evaluate postfix expression.<br>Algorithm for Postfix Evaluation – 05 Marks | 05 | CO2 | L3 |
| B. | Construct Binary Tree from given Inorder and Postorder traversals.<br>Let us consider the below traversals:<br>Inorder sequence: 4, 8, 2, 5, 1, 6, 3, 7<br>Postorder sequence: 8, 4, 5, 2, 6, 7, 3, 1<br>  <br>Left subtree – 03 Marks<br>Right subtree – 02 Marks | 05 | CO4 | L3 |
| C. | Write a C program to perform the following operations on a singly linked list: Insert at the end, Delete at the beginning<br><br>Insert at End – 03 marks<br><br>Delete at the beginning – 02 Marks | 05 | CO3 | L3 |
| D. | Explain all three cases of the delete operation in a Binary Search Tree with an example.<br>i. Delete a node having No Children – 01 Marks<br>ii. Delete a node having one Child – 02 Marks<br>iii. Delete a node having two Children – 02 Marks | 05 | CO4 | L3 |
| E. | Explain topological sorting with an example.<br>Explanation of topological sorting – 03 marks<br>Example – 02 Marks | 05 | CO5 | L2 |

| | | | | |
|---|---|---|---|---|
| F. | Explain the following Hash Functions with an example: division method, multiplication, mid-square, and folding.<br>Each method, along with an example and formula, carries 01 mark | 05 | CO6 | L2 |
| | *********** All the Best*********** | | | |