

Enrollment No./Seat No.:

GUJARAT TECHNOLOGICAL UNIVERSITY
PGDDS - SEMESTER - I EXAMINATION - WINTER 2025

Subject Code: DS01080011

Date: 18-12-2025

Subject Name: Data Structures and Algorithm

Time: 10:30 AM TO 01:00 PM

Total Marks: 70

Instructions

- 1. Attempt all questions.**
- 2. Make suitable assumptions wherever necessary.**
- 3. Figures to the right indicate full marks.**
- 4. Use of simple calculators and non-programmable scientific calculators are permitted.**

	Marks
Q.1 (a) Explain the fundamental difference between linear and non-linear data structures, providing one example for each.	03
(b) Explain the concept of Stack Overflow in the context of recursion. What is the cause of this error, and why is this risk generally absent when the same logic is implemented using Iteration (loops)?	04
(c) Briefly distinguish between the structure of a Singly Linked List and a Doubly Linked List based on the number and direction of the pointers in their nodes. Also Explain why insertion at the beginning of a Singly Linked List is highly efficient compared to insertion at a specific position in the middle of the list.	07
Q.2 (a) Distinguish between Static and Dynamic data structures based on their memory allocation. Give a common example for each type.	03
(b) Explain the primary advantage of a Circular Queue over a standard linear queue, and describe a scenario in a computer system where this structure is critically useful.	04
(c) Identify the two primary operations that define the behavior of a stack and specify the end of the stack where both of these operations occur. If the following sequence of operations is performed on an initially empty stack: PUSH(10), PUSH(20), POP, PUSH(30), POP, PUSH(40), what is the final state of the stack (list the elements from top to bottom)? Also explain the primary role of a stack when converting an Infix Expression to a Postfix Expression.	07
OR	
(c) Describe the two fundamental characteristics of a queue that differentiate it from a stack. In the context of an Operating System's CPU Scheduling, explain why a simple Queue is the most appropriate data structure to manage processes waiting for the CPU, and how the Dequeue operation specifically facilitates task execution.	07
Q.3 (a) Briefly explain the essential role played by the Stack in solving Tower of Hanoi problem.	03
(b) Explain the fundamental purpose of the rotation operation in an AVL Tree, and briefly describe one of the simplest rotation cases, the Single Rotation (LL or RR), used to restore the balance property after a violation.	04

- (c) For binary tree, define Root, Leaf Node, and Level of a Node. Also explain the difference between the Height (or Depth) of a Binary Tree and the Degree of a Node. Furthermore, what is the maximum possible degree of any node in a strictly defined Binary Tree? 07

OR

- (a) Briefly explain the two core rules that govern how a Stack is used to process and calculate the final result of a postfix expression. 03
- (b) Briefly explain the fundamental difference in the search strategy used by (i) Breadth-First Search and (ii) Depth-First Search. Also state the primary auxiliary data structure required to implement each method efficiently. 04
- (c) Given the following sequence of elements to be inserted into an initially empty BST: {60, 40, 75, 20, 50, 65, 90}. Draw the resulting Binary Search Tree, clearly showing the position of all seven nodes. Perform and state the resulting sequence of nodes for the following two standard traversals on the BST constructed: Inorder Traversal , Postorder Traversal 07

- Q.4** (a) Briefly describe the two main properties—Greedy Choice Property and Optimal Substructure—that an optimization problem must possess for a greedy algorithm to guarantee a globally correct, optimal solution. 03
- (b) Describe the specific comparison and exchange operation that occurs during a single pass of the Bubble Sort algorithm. What condition must the algorithm check to determine when the array is fully sorted, 04
- (c) Define a Spanning Tree. State the three essential properties that define a Minimum Spanning Tree (MST) specifically. Compare and contrast Prim's Algorithm and Kruskal's Algorithm for finding an MST based on their underlying strategy and the data structure each typically relies on to manage the selection of edges or vertices. 07

OR

- (a) Briefly describe the specific greedy choice made at each step to solve Activity Selection problem optimally, and explain which key parameter of the activities is used to guide this choice. 03
- (b) Explain the core principle of searching mechanism used in Binary Search that enables it to achieve a logarithmic time complexity. What single, critical pre-condition must the data structure satisfy for Binary Search to function correctly? 04
- (c) Explain the primary difference between finding the shortest path in an unweighted graph and in a weighted graph. Describe the core greedy strategy of Dijkstra's algorithm and explain the crucial role played by a Priority Queue in ensuring the algorithm's efficiency. 07
- Q.5** (a) If you have a knapsack capacity of $W = 10$ kg and two items: Item A (Value: $V=60$, Weight: $W=5$ kg and Item B (Value: $V=50$, Weight: $W=10$ kg, briefly describe the greedy criterion used for selection and apply it to determine which item is chosen first and why. 03
- (b) Explain the fundamental principle of optimal substructure utilized by Floyd-Warshall's All-Pairs Shortest Path algorithm, and identify the key variable or parameter that defines the intermediate steps (or 'subproblems') during its execution. 04

- (c) Describe the primary goal of the Partitioning process in Quick Sort. Given the array of numbers: {85, 20, 60, 10, 75, 50, 95}, using the first element, 85, as the initial Pivot, demonstrate the step-by-step partitioning process until the pivot is placed in its correct, final position. Clearly show the array's contents after the final swap that places the pivot. **07**

OR

- (a) Describe the specific greedy choice made at each step during the Huffman tree construction process, and explain the motivation behind this choice in terms of achieving optimal data compression. **03**
- (b) Explain the fundamental principle of Optimal Substructure as it applies to Matrix Chain Multiplication problem. Based on this, clearly define the subproblem that is iteratively solved, and state the primary inefficiency issue that Dynamic Programming efficiently resolves. **04**
- (c) Describe the two fundamental, recursive steps of the Merge Sort process: the Divide phase and the Conquer phase. Demonstrate the Merge phase using the following two sorted sub-arrays: $L = [15, 45, 70]$ and $R = [20, 60, 90]$. Show the step-by-step comparisons and the resulting sequence of elements inserted into the final merged array until the process is complete. **07**
