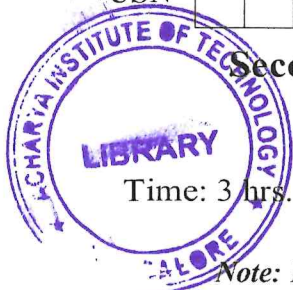


CBCS SCHEME

USN

--	--	--	--	--	--	--	--	--	--

MMC203



Second Semester MCA Degree Examination, June/July 2026

Data Structure and Algorithms

Time: 3 hrs.

Max. Marks: 100

- Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.
2. M : Marks, L: Bloom's level, C: Course outcomes.*

Module – 1			M	L	C
Q.1	a.	Define and explain the importance of data structures in computer science. What is an Abstract Data Type (ADT)? Discuss the relationship between ADTs and data structures with an example.	10	L2	CO1
	b.	Explain the algorithm analysis in terms of time and space complexity. Define Big O notation and explain its role with two examples.	10	L2	CO1
OR					
Q.2	a.	Write algorithms for inserting and deleting an element in a one-dimensional array. What are the time complexities of these operations? Also, explain any one real-world application of arrays.	10	L3	CO1
	b.	Differentiate between singly linked list, doubly linked list, and circular linked list in terms of structure, advantages, and typical applications. Also, write the steps to insert a node at the beginning in a singly linked list.	10	L4	CO1
Module – 2					
Q.3	a.	Define a stack. Describe the operations push, pop, and peek with an example. Explain how a stack is used in infix to postfix expression conversion with proper algorithm and one example.	10	L3	CO2
	b.	What is a queue? Write algorithms for enqueue and dequeue operations in a circular queue. Explain the advantage of a circular queue over a linear queue. Mention one application of each: priority queue and deque.	10	L3	CO2
OR					
Q.4	a.	Define recursion. Write a recursive algorithm for computing the factorial of a number. Also, write the recursive algorithm for Fibonacci series. Compare and analyze the time and space complexity of both algorithms.	10	L4	CO2

1 of 3

	b.	Write a recursive algorithm for the Tower of Hanoi problem. Solve it for 3 disks and explain each move. Derive the time complexity of the recursive algorithm.	10	L5	CO2
--	-----------	--	-----------	-----------	------------

Module – 3

Q.5	a.	Define a binary tree. Explain the terms root, leaf, internal node, height, and depth of a tree with an example diagram. Write and explain the algorithms for preorder, inorder, and postorder traversals of a binary tree.	10	L2	CO3
	b.	What is a Binary Search Tree (BST)? Write the algorithms for insertion and deletion in a BST. Illustrate your answer with an example showing creation, insertion, and deletion of nodes in a BST.	10	L3	CO3

OR

Q.6	a.	Define an AVL Tree. Explain the need for AVL trees. Show how the tree is rebalanced using LL, RR, LR, and RL rotations after insertions. Illustrate rotations with examples.	10	L4	CO3
	b.	Define a graph. Explain the terms vertex, edge, adjacency matrix, and adjacency list with suitable examples. Write algorithms for DFS and BFS traversal. Compare their working using an example graph.	10	L3	CO3

Module – 4

Q.7	a.	Explain the basic concepts and classification of sorting algorithms. Write the working of bubble sort and insertion sort algorithms with one example each. Compare their time complexities in best, average, and worst-case scenarios.	10	L2	CO4
	b.	Write and explain the algorithms for quicksort and illustrate the working of this algorithm with an example. Also, give its time complexities.	10	L3	CO4

OR

Q.8	a.	Differentiate between comparison-based sorting and non-comparison-based sorting algorithms. Explain the working of counting sort and radix sort with suitable examples.	10	L4	CO4
	b.	Write the algorithms for linear search and binary search with examples. Also explain hashing and two collision resolution techniques: chaining and open addressing, with brief examples.	10	L3	CO4

Module – 5

Q.9	a.	What is a Trie (prefix tree)? Explain its structure and working. How are tries used in dictionary and spell-checking applications? Write the algorithm for insertion and search operations in a trie with suitable examples.	10	L3	CO5
	b.	Define Segment Trees and Fenwick Trees (Binary Indexed Trees). Write algorithms for range sum queries and updates in both trees. Mention real-life use cases where these data structures are preferred.	10	L4	CO5
OR					
Q.10	a.	Explain the Divide and Conquer, Greedy, and Dynamic Programming (DP) algorithm design paradigms. Compare these with examples of problems solved using each method and mention their advantages and limitations.	10	L4	CO5
	b.	Discuss the importance of data structures and algorithms (DSA) in solving real-world problems. Give two detailed case studies where advanced DSA techniques were applied in industry applications. What best practices should be followed in implementing DSA in real-world systems?	10	L5	CO5

* * * * *