

25UCST301	DATA STRUCTURES AND ALGORITHMS	Category	L	T	P	R	Credit	Year of Introduction
		PCC	3	1	0	0	4	2025

Preamble: This course serves as a fundamental area of computer science that focuses on the structured organization, storage, and manipulation of data to develop efficient algorithmic solutions. It encompasses both linear and non-linear data structures, algorithm design strategies, and complexity analysis, equipping learners to assess performance trade-offs and construct optimized solutions for complex computational challenges.

Prerequisite: PIC204 Programming in C

Course Outcomes: After the completion of the course, the student will be able to:

CO	Course Outcomes	Knowledge Level
CO1	Apply algorithm complexity using asymptotic notations and implement linear data structures including arrays, stacks, and queues.	K3 Apply
CO2	Implement singly, doubly, and circular linked lists and apply suitable memory allocation strategies.	K3 Apply
CO3	Apply tree and graph data structures to perform traversals, searching, and real-world problem solving.	K3 Apply
CO4	Implement and compare sorting algorithms and hashing techniques with appropriate collision resolution strategies.	K3 Apply

Mapping of course outcomes with program outcomes and program-specific outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	3	2	-	-	-	-	-	-	-	2	3	2
CO2	3	3	2	-	-	-	-	-	-	-	2	3	2
CO3	3	3	3	2	-	-	-	-	-	-	2	3	2
CO4	3	3	2	2	-	-	-	-	-	-	3	3	3
Avg	3	3	2.25	2	-	-	-	-	-	-	2.25	3	2.25

Mapping of course outcomes with Knowledge and Attitude Profile (WK)

Wks Mapped with the course

	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO1	*	*	*	*	*			*	
CO2	*	*	*	*	*			*	
CO3	*	*	*	*	*			*	
CO4	*	*	*	*	*			*	
WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9	
*	*	*	*	*			*		

Mark Distribution:

Total Marks	CIE Marks	ESE Marks	ESE Duration
100	40	60	2.5 hrs

Continuous Internal Evaluation Pattern:

Attendance	: 5 marks
Continuous Assessment Test	: 20 marks
Continuous Assessment Assignment/ Seminar/CaseStudy	: 15 marks

SYLLABUS**Module 1 (14 hrs)**

Definitions; Data Abstraction; Performance Analysis - Time & Space Complexity, Asymptotic Notations; Polynomial representation using Arrays, Sparse matrix (Tuple representation); Stacks and Queues - Stacks, Queues, Circular Queues, Double Ended Queues, Priority Queue; Evaluation of Expressions- Infix to Postfix, Evaluating Postfix Expressions Searching Techniques: - Linear Search, Binary Search.

Module 2 (14 hrs)

Singly Linked List - Operations on Linked List, Stacks and Queues using Linked List, Polynomial representation using Linked List; Doubly Linked List; Circular Linked List; Memory allocation - First-fit, Best-fit, and Worst-fit allocation schemes.

Module 3 (14 hrs)

Trees:-Representation of Trees, Binary Trees-Types and Properties, Binary Tree Representation, Tree Operations, Tree Traversals; Expression Trees; Binary Search Trees - Binary Search Tree Operations; Binary Heaps - Binary Heap Operations. Graphs: - Definitions; Representation of Graphs; Depth First

Search and Breadth First Search; Applications of Graphs. Case Study: Design and Implementation of Tree and Graph Data Structures in a Smart City Management System. (Proof of Correctness needed)

Module 4 (14 hrs)

Sorting Techniques: - Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Bubble Sort, Heap Sort, Radix Sort. Case Study: Optimizing Data Processing in an E-Commerce System Using Appropriate Sorting Algorithms. Hashing – Hashing functions: Mid square, Division, Folding, Digit Analysis; Collision Resolution: Linear probing, Quadratic Probing, Double hashing, Open hashing.

Text Books

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms* (4th ed.). MIT Press.
2. Horowitz, E., Sahni, S., & Mehta, D. (2022). *Fundamentals of data structures in C* (6th ed.). Universities Press.
3. Tenenbaum, A. M., Langsam, Y., & Augenstein, M. J. (2021). *Data structures using C and algorithms* (2nd ed.). Prentice Hall.

Reference Books

1. Thareja, R. (2021). *Data structures using C* (3rd ed.). Oxford University Press.
2. Weiss, M. A. (2019). *Data structures and algorithm analysis in C* (4th ed.). Pearson.
3. Tremblay, J. P., & Sorenson, P. G. (2017). *Introduction to data structures with applications* (2nd ed.). Tata McGraw-Hill.

END SEMESTER EXAMINATION-MODEL QUESTION PAPER

Course Code	25UCST301
Course Name	DATA STRUCTURES AND ALGORITHMS
Max. Marks: 60	Duration: 2.5 Hours

Part A (Answer all questions. Each question carries 3 marks)				
Qn No.	Question	Marks	CO	BL
1.	Explain any two asymptotic notations used to express the complexity of the algorithm with the help of suitable examples.	3	1	L2
2.	Explain frequency count and Calculate the frequency count of the statement <pre>int fun(int n) { int i = 0, j = 0, k = 0, m = 0; i = n; for (i = 0; i < n; i++) for (j = 0; j < n; j++)</pre>	3	1	L3

	<pre> m += 1; for (i = 0; i < n; i++) for (k = 0; k < n; k++) m += 1; return m; } </pre>			
3.	A singly linked list contains the elements: $10 \rightarrow 20 \rightarrow 30 \rightarrow 40$ Apply the insertion operation to insert 25 after 20. Draw the resulting linked list.	3	2	L3
4.	A memory of size 500 KB contains free blocks of sizes: 100 KB, 50 KB, 200 KB, 150 KB Apply the First-Fit allocation strategy to allocate a process of size 120 KB. Show the allocation result.	3	2	L3
5.	Construct a Binary Search Tree (BST) by inserting the keys: 40, 20, 60, 10, 30, 50, 70 Perform the inorder traversal of the resulting tree.	3	3	L3
6.	Consider the graph with vertices {A, B, C, D, E} and edges: (A,B), (A,C), (B,D), (C,E) Apply Breadth First Search (BFS) starting from vertex A and list the order of traversal.	3	3	L3
7.	Apply Insertion Sort to sort the following list in ascending order: 35, 20, 45, 10, 30 Show the array after each pass.	3	4	L3
8.	A hash table of size 10 uses the hash function $h(k) = k \bmod 10$ Insert the keys 23, 43, 13, and 27 using Linear Probing for collision resolution. Show the final hash table.	3	4	L3
	Part B <i>(Answer any one full question from each Module. Each carry 9 marks)</i>			
	Module I			
Qn No.	Question	Marks	CO	BL
9	A sparse matrix is given below: $\begin{bmatrix} 0 & 0 & 5 & 0 & 0 \\ 0 & 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 & 0 \end{bmatrix}$ a) Represent the matrix using the Tuple Representation. (4 Marks) b) Determine the space required for storing the matrix in its original form and tuple form. (3 Marks)	9	1	L2

	c) Using asymptotic notation, compare the storage complexity of both representations. (2 Marks)													
	OR													
10	Consider the infix expression: (A+B)*(C-D)+E/F where A=6, B=4, C=8, D=3, E=12 and F=4. a) Convert the infix expression into postfix notation using a stack. (3 Marks) b) Evaluate the postfix expression showing the stack contents after each operation. (4 Marks) c) Apply Binary Search to locate the element 45 in the sorted array {10, 20, 30, 45, 50, 60, 75}. Show each step of the search process. (2 Marks)	9	1	L3										
	Module II													
Qn No.	Question	Marks	CO	BL										
11.A	A singly linked list contains the nodes: 15 → 25 → 35 → 45 → 55 a) Insert a new node containing 30 after node 25. (3 Marks) b) Delete the node containing 45 from the modified list. (3 Marks)	6	2	L3										
11.B	A doubly linked list contains: 10 ⇌ 20 ⇌ 30 ⇌ 40 Insert a node containing 25 between 20 and 30. Draw the resulting doubly linked list showing the links.	3	2	L2										
	OR													
12.A	The available memory blocks are: <table border="1"><thead><tr><th>Block</th><th>Size (KB)</th></tr></thead><tbody><tr><td>B1</td><td>100</td></tr><tr><td>B2</td><td>300</td></tr><tr><td>B3</td><td>200</td></tr><tr><td>B4</td><td>150</td></tr></tbody></table> Three processes require 120 KB, 90 KB, and 180 KB respectively. Apply the First-Fit memory allocation strategy and show the allocation after each request.	Block	Size (KB)	B1	100	B2	300	B3	200	B4	150	6	2	L3
Block	Size (KB)													
B1	100													
B2	300													
B3	200													
B4	150													
12.B	A circular linked list contains: 10 → 20 → 30 → 40 → (back to 10) Apply the operation to insert 50 after node 40 and draw the resulting circular linked list.	3	2	L2										
	Module III													
Qn No.	Question	Marks	CO	BL										
13.A	Compare graphs and trees in terms of their structure,	5	3	L2										

	usage, and properties, and explain each data structure is suitable for its respective application.			
13.B	A network of four cities A, B, C and D has the following direct roads: A–B, A–C, B–C and C–D. Using the same network, construct a graph representation in the form of an adjacency list. Show the list and matrices for each vertex	4	3	L3
	OR			
14.A	Demonstrate how Depth-First Search (DFS) and Breadth-First Search (BFS) algorithms explore a graph. Explain the step-by-step process each algorithm follows during traversal and highlight the key differences in their strategies and order of visiting nodes.	5	3	L3
14.B	Construct a Binary Search Tree for the following values 62, 14, 96, 12, 105, 3, 75, 22, 87, 32, 20, 13, 102, 68, 125. Write the output of inorder, preorder & postorder traversals on the tree.	4	3	L3
	Module IV			
Qn No.	Question	Marks	CO	BL
15.A	Given the values 45, 36, 19, 58 and 10 and a hash table of size 13. Show the resultant table after inserting values in the given order by using hash function $h(k) = k \% 13$ with each of the following collision strategies. i) Linear Probing ii) Quadratic Probing iii) Double hashing with secondary hash function $h'(k) = 7 - k \% 7$	5	4	L3
15.B	Create a Max-heap from 20, 12, 30, 25, 18, 8 and 24 using heapify. Then delete the root element from the tree.	4	4	L3
	OR			
16.A	Write an algorithm to sort an array using merge sort. Apply merge sort to sort the numbers 45, 36, 12, 65, 24, 11, 44, 54, 17 and 33.	5	4	L3
16.B	Perform radix sort to sort the numbers 341, 22, 33, 122, 42, 5 and 4.	4	4	L3