

R.N.G.PATEL INSTITUTE OF TECHNOLOGY-RNGPIT
(An Autonomous Institute U/s UGC Act 1956)

B.Tech. SEMESTER-III, SEMESTER END EXAMINATION – WINTER 2025

SUBJECT CODE: 1CS301

DATE: 16-12-2025

SUBJECT NAME: DATA STRUCTURES AND ALGORITHMS

TIME: 11:00 AM to 01:30 PM

TOTAL MARKS: 70

Instructions

1. It is **compulsory** for students to write **Enrolment No. /Seat No.** on the question paper.
2. Write answers of **Section A** and **Section B** in **separate answer books**.
3. Attempt all questions from both **Section A** and **Section B**.
4. Each section carries **35 marks**, with a total of **70 marks** for the examination.
5. The figures to the right of each question indicate full marks, make suitable assumptions with justification.
6. BL - Cognitive Level (As per Revised Bloom's Taxonomy) (R-Remember, U-Understanding, A –Application, N –Analyze, E – Evaluate, C -Create), CO - Course Outcomes.

SECTION A

		Marks	BL	CO
Q.1	(a) State primitive and non-primitive data structure and classify with two examples of each.	03	R	1
	(b) Explain row-major order and column-major order representation of 2-D array.	04	U	1
Q.2	(a) Enlist the applications of stack in computer science.	03	R	2
	(b) Write an algorithm for insertion of node at first position in linear linked list.	04	U	2
	(c) Convert the following infix expression to prefix expression using stack: $A - B / (C * D ^ E)$	07	A	2
OR				
Q.2	(a) Compare array and linked List.	03	R	2
	(b) State the advantage of Circular Queue over Queue. Write an algorithm to insert an element in Circular Queue.	04	U	2
	(c) Convert the following infix expression to postfix expression and evaluate it using stack : $9 - ((3 * 4) + 8) / 4$	07	A	2
Q.3	(a) Compare internal sorting and external sorting methods.	03	N	5
	(b) Write an algorithm of binary search.	04	U	5
	(c) Sort the following data using merge sort: 10, 15, 28, 09, 40, 35, 29, 12, 06, 07	07	A	5

OR

- | | | | | |
|------------|---|-----------|----------|----------|
| Q.3 | (a) Compare linear search and binary search methods. | 03 | N | 5 |
| | (b) Write an algorithm of bubble sort. | 04 | U | 5 |
| | (c) Sort the following data using quick sort :
65, 70, 75, 80, 85, 60, 55, 50, 45 | 07 | A | 5 |

SECTION B

		Marks	BL	CO
Q.4	(a) Define file and explain any three file attributes maintained by an operating system.	03	U	6
	(b) Describe how file indexing is implemented. Discuss primary and secondary indexes with neat diagrams.	04	U	6
Q.5	(a) Differentiate between open addressing and chaining.	03	U	3
	(b) Explain the division, mid-square, and folding methods of hashing functions with examples.	04	A	3
	(c) A hash table of size 11 uses $h(k) = k \text{ mod } 11$. Insert the keys: 21, 32, 43, 54, 65, 76 using Quadratic Probing. Show the probing sequence and final table.	07	N	3
OR				
Q.5	(a) Define load factor. How does it affect hash table performance?	03	U	3
	(b) Describe the collision resolution techniques: linear probing, quadratic probing, and double hashing.	04	A	3
	(c) A hash table has $m = 13$. Use the following hash functions: $h_1(k) = k \text{ mod } 13$, $h_2(k) = 7 - (k \text{ mod } 7)$. Insert keys 18, 41, 22, 44, 59, 32, 31, 73 using Double Hashing.	07	E	3
Q.6	(a) Define the following terms: degree of a node, leaf node, height of a tree.	03	R	4
	(b) Given the preorder and inorder traversal sequences below, construct the binary tree. Preorder: A, B, D, E, C, F Inorder: D, B, E, A, C, F	04	N	4
	(c) Evaluate the performance of the Binary Search Tree structures for insertion, deletion, and searching operations. Support your answer with an example and a time complexity table.	07	E	4
OR				
Q.6	(a) Construct the binary search tree (BST) for the sequence of numbers: 40, 20, 10, 30, 60, 50, 70. Show the structure of the resulting tree.	03	A	4

- (b) Use Kruskal's algorithm to find the minimum spanning tree (MST) for the given graph: **04 N 4**

Edge	Weight
A-B	3
A-C	1
B-C	7
B-D	5
C-D	2

- (c) Construct an expression tree for the postfix expression: **07 E 4**
 $AB+C*DE-/$
Evaluate the expression if $A=5, B=3, C=2, D=8, E=4$.
Show step-by-step computation.
