

 SIES Graduate School of Technology RISE WITH EDUCATION NMACA+ (Affiliated to University of Mumbai)	End Semester Examination (R-24) SH 2025 Answer Key with marking scheme
Branch: IT	Course: Data Structure and Analysis
Year/ Semester: SE/III	Course code: ITC302
Time: 03 hours	Marks: 80

Q.1 - Attempt any FOUR

A. Static vs Dynamic Data Structures (5 marks)

Parameter	Static	Dynamic	Marks
Memory	Fixed size at compile time	Runtime allocation	1
Examples	Arrays	Linked lists, trees	1
Flexibility	Less flexible	More flexible	1
Memory Usage	Efficient if size known	Efficient for unknown size	1
Access	Random access	Sequential access	1

B. Types of Binary Trees (5 marks)

Type	Definition	Properties	Marks
Full	0 or 2 children per node	All levels except last filled	1.5
Complete	All levels filled, last left to right	Height balanced	1.5
Perfect	All internal nodes have 2 children	$2^h - 1$ nodes	1
Balanced	Height difference ≤ 1	Efficient operations	1

C. Adjacency Matrix (5 marks)

Component	Description	Marks
Definition	$V \times V$ matrix, $A[i][j] = 1$ if edge exists	1
Properties	Symmetric for undirected graphs	1
Space	$O(V^2)$	1
Operations	$O(1)$ edge check	1
Use Case	Dense graphs	1

D. Garbage Collection (5 marks)

Aspect	Description	Marks
Definition	Automatic memory reclamation	1
Importance	Prevents memory leaks	1
Types	Reference counting, mark-sweep	2
Benefits	Program reliability	1

E. Mid-Square Hash Function (5 marks)

Example: Key = 1234

1. Square: $1234^2 = 1522756$

2. Extract middle digits: 227

3. Hash: $227 \% \text{table_size}$

Step	Description	Marks
Concept	Square key, take middle digits	2
Example	Step-by-step calculation	2

Advantages	Uniform distribution	1
------------	----------------------	---

F. Stack in Expression Evaluation (5 marks)

Role	Description	Marks
Operator storage	Hold operators until operands available	1
Precedence	Manage operator hierarchy	1
Parenthesis	Handle nested expressions	1
Conversion	Infix to postfix	1
Evaluation	Postfix calculation	1

Q.2 - Attempt any FOUR

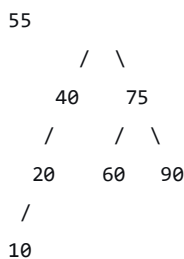
A. BST Operations (10 marks)

Insert: 55, 30, 75, 20, 40, 60, 90, 10

Delete 30: Replace with inorder successor (40)

Final Tree:

text



Operation	Description	Marks
Insertion	Correct BST construction	4
Deletion	Proper node removal	4
Final tree	Valid BST structure	2

B. Prim's Algorithm (10 marks)

Steps:

1. Start from any vertex
2. Add minimum weight edge connecting to tree
3. Repeat until all vertices included

Step	Description	Marks
Algorithm	Correct Prim's implementation	3
Edge selection	Minimum weight edges	3
MST construction	Step-by-step process	2
Total weight	Final cost calculation	2

C. Quick Sort (10 marks)

Array: [170, 45, 75, 90, 2, 802, 24, 66]

Steps:

1. Pivot 170: [45,75,90,2,24,66] 170 [802]
2. Left: Pivot 45: [2,24] 45 [75,90,66]

3. Final: [2,24,45,66,75,90,170,802]

Step	Description	Marks
Partitioning	Correct pivot placement	4
Recursive calls	Left and right sorting	3
Final array	Sorted order	2
Complexity	O(n log n) average	1

D. Circular Queue using Array (10 marks)

```
#define SIZE 100
int queue[SIZE];
int front = -1, rear = -1;

void enqueue(int value) {
    if ((rear + 1) % SIZE == front) {
        printf("Queue Full");
        return;
    }
    if (front == -1) front = 0;
    rear = (rear + 1) % SIZE;
    queue[rear] = value;
}

int dequeue() {
    if (front == -1) return -1;
    int value = queue[front];
    if (front == rear) front = rear = -1;
    else front = (front + 1) % SIZE;
    return value;
}
```

Component	Marks
Circular increment	3
Full/empty conditions	3
Enqueue operation	2
Dequeue operation	2

E. Threaded Binary Tree (10 marks)

Aspect	Description	Marks
Definition	Nodes contain threads to predecessors/successors	2
Types	Single/double threaded, left/right threads	3
In-order Successor	Right child or thread to next node	3
Advantages	No stack, efficient traversal	2

F. Huffman Coding (10 marks)

Frequencies: A(15), B(7), C(6), D(6), E(5), F(4), G(3), H(2), I(2)

Steps:

1. Combine H(2)+I(2)=4

2. Combine G(3)+F(4)=7

3. Continue merging...

Codes: A:11, B:101, C:001, D:000, E:011, F:1001, G:1000, H:0100, I:0101

Compression:

Huffman: 146 bits

ASCII: 400 bits

Ratio: $400/146 \approx 2.74$

Component	Marks
Tree construction	4
Code assignment	2
Bit calculation	2
Compression ratio	2

Q.3 - Attempt any FOUR

A. Binary vs Linear Search (5 marks)

Parameter	Binary Search	Linear Search	Marks
Prerequisite	Sorted array	Any array	1
Time	$O(\log n)$	$O(n)$	1
Space	$O(1)$	$O(1)$	1
Best Case	$O(1)$	$O(1)$	1
Use Case	Large sorted data	Small data	1

B. Stack Applications (5 marks)

Application	Description	Example	Marks
Function Calls	Call stack management	Recursion	1.5
Expression Evaluation	Infix to postfix	Calculator	1.5
Undo/Redo	Operation history	Text editor	1
Backtracking	Path finding	Maze solving	1

C. Priority Queue (5 marks)

Component	Description	Marks
Definition	Elements with priorities	1
Operations	Insert, extract-max/min	1
Implementation	Heap, linked list	1
Application	Dijkstra's algorithm	1
Example	CPU scheduling	1

D. Recursive Control Flow (5 marks)

Phase	Description	Marks
Winding	Function calls itself	2
Base Case	Termination condition	1
Unwinding	Return from calls	2

E. Linear vs Nonlinear DS (5 marks)

Parameter	Linear	Nonlinear	Marks
Arrangement	Sequential	Hierarchical	1
Traversal	Single path	Multiple paths	1

Examples	Array, list	Tree, graph	1
Memory	Contiguous	Non-contiguous	1
Complexity	Simple	Complex	1

F. Graph in Social Networks (5 marks)

Application	Description	Marks
Friend Network	Vertices=users, edges=friendships	1
Recommendations	Suggest friends	1
Community Detection	Find groups	1
Influence Analysis	Key influencers	1
Information Spread	News/viral content	1