

MODULE 1: INTRODUCTION TO DATA STRUCTURES

- ✚ Definition and types of data structures
- ✚ Static vs Dynamic structures
- ✚ Abstract Data Types (ADT)
- ✚ Time & Space Complexity
- ✚ Recursion vs Iteration

MODULE 2: ARRAYS AND STRINGS

- ✚ 1D and 2D Arrays
- ✚ Operations: Traversal, Insertion, Deletion, Search
- ✚ Sorting:
 - Bubble, Selection, Insertion Sort
 - Merge Sort, Quick Sort
- ✚ Searching:
 - Linear Search, Binary Search
- ✚ Strings:
 - String operations
 - String manipulation techniques
 - Pattern matching: KMP, Rabin-Karp

MODULE 3: LINKED LISTS

- ✚ Singly Linked List
- ✚ Doubly Linked List
- ✚ Circular Linked List
- ✚ Operations: Insertion, Deletion, Traversal
- ✚ Applications (e.g., polynomial addition, LRU cache)

MODULE 4: STACKS

- ✚ Definition and operations
- ✚ Implementation using array and linked list
- ✚ Infix, Prefix, Postfix expressions
- ✚ Applications:
 - Expression evaluation
 - Syntax parsing
 - Undo feature in editors

MODULE 5: QUEUES

- ✚ Linear Queue
- ✚ Circular Queue
- ✚ Deque (Double-Ended Queue)
- ✚ Priority Queue
- ✚ Implementation (Array/Linked List)
- ✚ Applications: Scheduling, buffers

MODULE 6: TREES

- ✚ Binary Tree
- ✚ Binary Search Tree (BST)
- ✚ AVL Tree (Self-balancing)
- ✚ B-Trees, B+ Trees
- ✚ Tree Traversals: In-order, Pre-order, Post-order, Level-order
- ✚ Applications: Expression Trees, Decision Trees

MODULE 7: HEAPS

- ✚ Max Heap, Min Heap
- ✚ Heap Operations (Insert, Delete, Heapify)
- ✚ Heap Sort
- ✚ Applications: Priority Queues, Scheduling

MODULE 8: GRAPHS

- ✚ Representations: Adjacency Matrix/List
- ✚ Types: Directed, Undirected, Weighted
- ✚ Graph Traversals:
 - BFS (Breadth First Search)
 - DFS (Depth First Search)
- ✚ Shortest Path Algorithms:
 - Dijkstra, Bellman-Ford
 - Floyd-Warshall, A*
- ✚ Minimum Spanning Tree:
 - Kruskal's, Prim's Algorithm
- ✚ Applications: Maps, Social Networks, AI pathfinding

MODULE 9: HASHING


- ✚ Hash Functions
- ✚ Collision Handling: Chaining, Open Addressing
- ✚ Hash Table implementation
- ✚ Applications: Indexing, Caching

MODULE 10: ADVANCED TOPICS & REAL-WORLD APPLICATIONS

- ✚ Trie (Prefix Tree)
- ✚ Segment Tree
- ✚ Fenwick Tree (Binary Indexed Tree)
- ✚ Disjoint Set (Union-Find)
- ✚ LRU Cache, Top K elements
- ✚ Real-world problems using data structures (LeetCode, GFG)

PRACTICE & MINI PROJECTS

- ✚ Live coding: Daily DSA problems
- ✚ Competitive programming rounds
- ✚ Mini Projects:
 - Spell checker using Trie
 - Movie Recommendation System using Graphs
- ✚ Memory Management Simulator (Stack + Heap)

 **Assessment:** Weekly coding tests, quizzes, and one final capstone project.