DSA Syllabus

1. Introduction

- What is DSA?
- Importance in programming
- Recursion basics

2. Time & Space Complexity

- Time Complexity
- Space Complexity
- Big-O, Big-Theta, Big-Omega notation

3. Arrays

- 1D and 2D arrays
- Operations: traversal, insertion, deletion
- Prefix sum, subarrays

4. Strings

- String basics and operations
- String manipulation and searching

5. Linked List

- Singly and doubly linked list
- Operations: insertion, deletion, traversal
- Circular linked list

6. Stack

- Concept and implementation (array/linked list)
- Applications

7. Queue

- Concept and implementation
- Circular queue, deque, priority queue

8. Hashing

- Hash tables, maps, sets
- Collision handling

9. Recursion & Backtracking

- Recursion basics
- Backtracking concepts

10. Trees

- Binary tree, binary search tree
- Tree traversal techniques
- Advanced trees: AVL, Trie

11. Graphs

- Representation: adjacency matrix/list
- BFS and DFS
- Graph algorithms: shortest path, MST, cycle detection, topological sort

12. Searching & Sorting

- Linear search, binary search
- Bubble, selection, insertion, merge, quick, heap sort
- Advanced sorting techniques

13. Heaps & Priority Queue

- Min-heap, max-heap
- Heap operations

14. Dynamic Programming (DP)

- Memoization and tabulation
- Basic DP concepts

15. Greedy Algorithms

- Greedy approach basics
- Applications

16. Miscellaneous

- Bit manipulation
- Sliding window technique
- Two pointer technique
- Union-Find / Disjoint Set
- Segment Tree / Fenwick Tree