Core DSA Patterns You Need for Interviews

This list focuses on the minimum number of patterns required to crack software engineering interviews efficiently.

Array & String Patterns

- Two Pointers
- Sliding Window (Fixed & Variable)
- Prefix Sum
- Difference Array / Range Updates
- Kadane’s Algorithm (Maximum Subarray)

Hashing & Counting

- Frequency Map / HashMap Logic
- Subarray Sum = K (Prefix Sum + HashMap)

Binary Search Patterns

- Standard Binary Search
- Binary Search on Answer
- Lower Bound / Upper Bound / First & Last Occurrence

Recursion & Divide and Conquer

- Recursion & Backtracking
- Divide & Conquer (Merge Sort, Inversion Count)

Linked List Patterns

- Fast & Slow Pointer Technique
- Reverse & Merge Linked Lists

Stack Patterns

- Monotonic Stack
- Histogram / Expression Evaluation Problems

Heap Patterns

- Top K / Kth Element
- Two Heaps Technique

Tree & Graph Patterns

- DFS & BFS
- Binary Tree Traversals
- Lowest Common Ancestor (LCA)
- Graph Traversal & Connected Components

Dynamic Programming Patterns

- 1D Dynamic Programming
- 2D Dynamic Programming
- Knapsack Pattern
- LCS / LIS Pattern