DSA Roadmap (Beginner → Advanced)

1. Foundations (Basics of Programming & Math)

- Time & Space Complexity (Big-O, Big-Ω, Big-Θ)
- Recursion & Backtracking basics
- Basic Math for CP (prime numbers, GCD, LCM, modulo, power, combinatorics)
- Resources: CLRS, Abdul Bari, HackerRank Warmup

2. Arrays & Strings (Easy Level DSA)

- Arrays basics: traversal, insertion, deletion
- Prefix Sum, Sliding Window, Two Pointers
- Strings: Palindrome, Substrings, Anagrams
- Hashing (frequency map, sets)
- Problems: Reverse array, Kadane’s Algorithm, Longest Substring w/o Repeat, Group Anagrams

3. Sorting & Searching

- Sorting: Bubble, Selection, Insertion, Merge Sort, Quick Sort, Counting Sort
- Searching: Binary Search (and variations)
- Binary Search on Answer (Allocate Books, Aggressive Cows)
- Problems: Search in Rotated Array, Median of Two Sorted Arrays, Kth Largest Element

4. Linked List

- Singly Linked List (reverse, middle, cycle detection)
- Doubly Linked List
- Merge Two Sorted Linked Lists
- LRU Cache (LL + HashMap)

5. Stack & Queue

- Stack (Next Greater Element, Balanced Parentheses, Min Stack)
- Queue basics, Circular Queue
- Deque (sliding window maximum)
- Monotonic Stack/Queue

6. Recursion & Backtracking

- Subsets / Subsequence generation
- N-Queens Problem, Rat in a Maze
- Word Search, Sudoku Solver

7. Trees & Binary Search Trees

- Traversals (Inorder, Preorder, Postorder, Level Order)

- Height, Diameter of Tree, LCA
- BST (insert, delete, search, validate)
- Balanced BST (AVL, Red-Black basics)

8. Heaps & Priority Queue

- Min Heap, Max Heap, Heap Sort
- Kth Largest/Smallest Element
- Merge K Sorted Lists
- Top K Frequent Elements

9. Hashing & Maps

- HashMap, HashSet, Unordered Map
- Problems on Hashing (Subarray with Sum K, Count Distinct Elements in Window)
- Trie (Word Search, Autocomplete system)

10. Graphs

- Graph Representation (Adj List, Matrix)
- BFS & DFS
- Shortest Path (Dijkstra, Bellman-Ford, Floyd-Warshall)
- MST (Kruskal, Prim), Topological Sort
- Disjoint Set Union (Union Find)
- Problems: Number of Islands, Word Ladder, Clone Graph

11. Dynamic Programming

- Basics: Memoization vs Tabulation
- 1D DP: Fibonacci, Climbing Stairs, House Robber
- 2D DP: Knapsack, Subset Sum, Matrix Path
- String DP: LCS, Edit Distance, Palindrome Partitioning
- Advanced: DP on Trees/Graphs, Digit DP, Bitmask DP
- Problems: LIS, Coin Change, Equal Subset Sum, Max Path Sum

12. Advanced Topics

- Segment Trees & Fenwick Tree (BIT)
- DSU + Path Compression
- Range Queries (RMQ, Lazy Propagation)
- Advanced Graph (Max Flow, Min Cut, Bipartite Matching)
- Computational Geometry (Convex Hull, Line Sweep)

Practice Strategy

- Solve 2–3 problems daily (mix easy + medium, 1 hard weekly)
- Platforms: LeetCode, Codeforces/AtCoder, GeeksforGeeks
- Maintain notes / GitHub repo for formulas & patterns
- Revise problem-solving patterns (sliding window, greedy, DP states)

Suggested Timeline (6 Months)

- Month 1–2 → Basics, Arrays, Strings, Recursion, Sorting, Searching, Linked List
- Month 3 → Stack, Queue, Trees, Hashing
- Month 4 → Graphs + DP Basics
- Month 5 → Advanced DP + Heaps + Tries
- Month 6 → Segment Trees, Advanced Graph, Mock Interviews