DSA with Resources

Data Structures And Algorithms:

ROADMAP:

🚀 DSA Learning Roadmap

Step 1: Foundation (Mathematics & Complexity Analysis)

- Big-O Notation (Time & Space Complexity)
- Recursion (Base cases, Stack Memory, Tail Recursion)
- Mathematical Proofs (Induction, Pigeonhole Principle)
- Bit Manipulation (XOR tricks, Bitmasking)

Step 2: Basic Data Structures

1️⃣ Arrays & Strings

- Traversal, Searching, Sorting
- Sliding Window, Two Pointers
- Prefix Sum, Kadane's Algorithm
- String Matching (KMP, Rabin-Karp)

2️⃣ Linked Lists (Singly, Doubly, Circular)

- Reversal, Merging, Detecting Cycles (Floyd's Cycle Detection)
- Fast & Slow Pointer approach

3️⃣ Stacks & Queues

- Implementation (Array, Linked List)
- Monotonic Stack/Queue
- Min Stack, Next Greater Element

Step 3: Recursion & Backtracking

- Permutations & Combinations
- Subsets, Subsequence Generation
- N-Queens, Sudoku Solver
- Rat in a Maze, Word Search

Step 4: Sorting & Searching

1️⃣ Sorting Algorithms

- QuickSort, MergeSort, HeapSort
- Counting Sort, Radix Sort

2️⃣ Searching Algorithms

- Binary Search (Lower & Upper Bound)
- Ternary Search
- Order Statistics (Kth Largest/Smallest)

Step 5: Hashing & Maps

- HashMap, HashSet
- Collision Handling
- Rolling Hash, Anagram Grouping

Step 6: Trees & Graphs

1 Trees (BST, AVL, Segment Tree, Trie)

- DFS, BFS Traversals
- Lowest Common Ancestor (LCA)
- Binary Tree Views (Top, Bottom, Left, Right)

2 Graphs (DFS, BFS, Dijkstra, MST)

- Graph Representation (Adjacency List, Matrix)
- Shortest Path (Dijkstra, Bellman-Ford)
- Topological Sorting (Kahn's Algorithm, DFS-based)
- Disjoint Set Union (Union-Find)
- Bridges & Articulation Points

Step 7: Dynamic Programming (DP)

- 1D DP: Fibonacci, Climbing Stairs, Coin Change
- 2D DP: Knapsack, Grid Paths, Longest Common Subsequence
- DP with Bitmasking

- DP on Trees & Graphs

Step 8: Advanced Topics (For Competitive Programming)

- Segment Tree, Fenwick Tree
- Heavy Light Decomposition
- Square Root Decomposition
- Game Theory
- Number Theory & Combinatorics

🎯 Practice Strategy

1. Easy Problems (50-100 questions) → Build confidence
2. Medium Problems (200+ questions) → Master standard patterns
3. Hard Problems (100+ questions) → Improve problem-solving

🔹 Platforms: LeetCode, Codeforces, CodeChef, AtCoder, GFG
🔹 Competitions: Participate in Codeforces Div2/Div3, Leetcode contests

🎯 Final Advice

🔹 For 7-10 LPA: Master LeetCode Mediums, do 200-500 DSA problems
🔹 For 25+ LPA: Deep dive into Graphs, DP, Advanced DSA, compete in Codeforces Div1
🔹 For 40+ LPA: Become a competitive programming expert, master mathematical optimization

YOUTUBE CHANNELS:

English channel:
https://youtube.com/playlist?list=PLgUwDviBIf0oF6QL8m22w1hIDC1vJ_BHz&si=7Xr_TvxWvg0gai-6

Telugu Channel:
https://youtube.com/playlist?list=PLjzLBp9HHZWhVXBSPS1VqxXXDoVk07gd9&si=1Nr6mZbKc-jUuffE

Hindi Channel:
https://youtube.com/playlist?list=PLDzeHZWIZsTryvtXdMr6rPh4IDexB5NIA&si=mHmo872okeDvKqK8

ALL THE BEST