⭐ THE COMPLETE DSA ROADMAP (BEGINNER → ADVANCED)

(Includes all important topics + subtopics in the correct sequence)

🟦 PHASE 1 — Foundations (Week 1–2)

1️⃣Arrays (Start here)

- Basics: indexing, iteration
- Insertion, deletion
- Searching in arrays
- Prefix Sum
- Kadane's Algorithm
- Sliding Window
- **Two-Pointer Technique** (subset of arrays)
    - Pair sum / triplet sum
    - Remove duplicates
    - Move zeroes
    - Reverse array
    - Dutch National Flag problem (sort colors)

2️⃣Strings

- String traversal
- ASCII & character mapping

- Palindrome check
- Anagrams
- String sorting
- Frequency count
- Two-pointer problems on strings
- Pattern matching basics

3️⃣STL in C++ (Start here itself)

Use from Day 1, parallel to all topics.

STL Containers:

- vector
- pair
- stack
- queue
- deque
- priority_queue
- set
- unordered_set
- map
- unordered_map

Algorithms:

- sort(), stable_sort()
- reverse()
- lower_bound(), upper_bound()

- accumulate()
- max_element(), min_element()

Iterators:

- begin(), end()
- auto keyword
- range-based for loops

🟩 PHASE 2 — Linear Data Structures (Week 3–4)

4️⃣Linked List

- Singly linked list
- Insert, delete operations
- Fast & slow pointer method
- Reverse linked list
- Detect cycle (Floyd's cycle detection)
- Middle of LL
- Merge two sorted LL
- Remove nth node from end

5️⃣Stack

- Stack using array
- Stack using linked list

- Applications:
    - Valid parentheses
    - Next greater element
    - Infix → postfix
    - Stock span

6️⃣Queue

- Queue using array
- Circular queue
- Queue using two stacks
- Priority queue
- Deque
- Sliding window max using deque

🟧 PHASE 3 — Algorithm Basics (Week 5–6)

7️⃣Searching

- Linear search
- Binary search
- Binary search on answers (VERY important)
    - Aggressive cows
    - Allocate books
    - Koko eating bananas

8️⃣Sorting

- Bubble, Insertion, Selection
- Merge sort
- Quick sort
- Heap sort (optional)
- Counting sort (concept)
- Custom comparator (STL)

9️⃣Recursion

- Base case
- Recursive tree
- Factorial, Fibonacci
- Sum of array
- Reverse array via recursion
- Subsequence generation
- Recursion with strings
- Recursion with backtracking basics

🟨 PHASE 4 — Greedy Algorithms (Week 7)

🔟 Greedy Algorithm Topics

- Greedy choice property
- Activity selection
- Fractional knapsack

- Job sequencing
- Minimum meeting rooms
- Interval scheduling problems
- Gas station
- Jump game
- Huffman Coding
- Minimum coin change (greedy version)

🟥 PHASE 5 — Backtracking (Week 8)

- N-Queens
- Sudoku solver
- Subsets
- Permutations
- Combination sum
- Rat in a maze
- Word search
- Palindrome partitioning

🟦 PHASE 6 — Trees (Week 9–10)

Binary Trees

- Node structure
- DFS traversals

    - Inorder, Preorder, Postorder
- BFS (level order traversal)
- Height, depth
- Diameter of tree
- Balanced tree
- Sum tree
- Boundary traversal

Binary Search Trees (BST)

- Insertion
- Searching
- Deletion
- Validate BST
- Kth smallest/largest

Advanced Trees

- AVL trees (concept)
- Segment tree (optional)
- Trie (important for strings)

🟩 PHASE 7 — Graphs (Week 11–12)

Graph Basics

- Adjacency list
- BFS

- DFS
- Connected components
- Cycle detection (directed + undirected)
- Bipartite graph

Shortest Paths

- Dijkstra (Greedy + Priority Queue)
- Bellman-Ford
- Floyd–Warshall

Minimum Spanning Trees (Greedy)

- Prim’s algorithm
- Kruskal’s algorithm (with DSU/Union-Find)

Topological Sort

- Kahn’s algorithm
- DFS-based topo sort

🟪 PHASE 8 — Dynamic Programming (Week 13–16)

Beginner Level

- Fibonacci
- Climbing stairs
- Grid path (2D DP)
- Subset sum

- 0/1 Knapsack

Intermediate

- LIS
- LCS
- Edit distance
- Matrix chain multiplication

Advanced

- DP on Trees
- DP on Graphs
- Bitmask DP

⭐ FINAL SUMMARY — COMPLETE TOPIC ORDER

1. Arrays
2. Strings
3. 2-Pointer
4. STL
5. Linked List
6. Stack
7. Queue
8. Searching
9. Sorting

10. Recursion

11. Greedy

12. Backtracking

13. Trees

14. BST

15. Graphs

16. DP