DSA (Data Structures and Algorithms) Syllabus – Hinglish Version

📘 1. DSA Introduction

- DSA kya hota hai?
- DSA ka use kyu important hai (coding interviews ke liye especially)
- Time Complexity (Big O notation samajhna)
- Space Complexity
- Code ka performance kaise analyze karte hain
- Big O, Big Ω, Big Θ Notations

🧩 *Example:*
Loop ka time complexity check karna (like O(n), O(n²), etc.)

📊 2. Basic Data Structures

🔷 Array

- Array kya hota hai aur kaise kaam karta hai
- Fixed size vs Dynamic array
- Operations: Insert, Delete, Search, Traverse
- Multi-dimensional arrays (2D arrays)
- Sliding window concept

💡 *Practice:*
Reverse array, Rotate array, Maximum subarray sum (Kadane's Algo)

🔷 String

- String basics (characters ka sequence)
- String operations: compare, reverse, substring
- Palindrome check karna
- String pattern matching (KMP, Rabin-Karp algorithms)

💡 *Practice:*
Check Anagram, Longest substring without repeating characters

---

🔷 Linked List

- Linked List kya hoti hai (Node-based structure)
- Types:
  - Singly Linked List
  - Doubly Linked List
  - Circular Linked List
- Operations: Insert, Delete, Traverse, Reverse

💡 *Practice:*
Detect Loop in Linked List (Floyd's Algorithm)

🧱 3. Stack aur Queue

🟩 Stack

- LIFO (Last In, First Out) principle
- Stack implementation (array / linked list se)
- Uses: Expression evaluation, Parentheses check

💡 *Practice:*
Next Greater Element, Postfix Evaluation

🟦 Queue

- FIFO (First In, First Out) principle
- Normal, Circular, Priority aur Deque (Double Ended Queue)
- Queue implementation (array / linked list se)

💡 *Practice:*
Implement Queue using two Stacks

🌱 4. Recursion

- Function calling itself
- Base case aur recursive case samajhna
- Recurrence relation banana
- Backtracking ka concept

💡 *Practice:*
Factorial, Fibonacci, Tower of Hanoi, N-Queens Problem

🌳 5. Trees

- Tree kya hota hai (hierarchical structure)
- Binary Tree aur Binary Search Tree (BST)
- Traversals: Inorder, Preorder, Postorder
- Balanced Trees (AVL, Red-Black Tree intro)
- Binary Heap (Min aur Max Heap)

💡 *Practice:*
Insert/Delete in BST, Lowest Common Ancestor (LCA)

🌐 6. Graphs

- Graph kya hota hai (nodes + edges)
- Directed / Undirected graphs
- Graph representation (Adjacency List / Matrix)
- BFS (Breadth First Search), DFS (Depth First Search)
- Shortest Path: Dijkstra, Bellman-Ford
- MST: Kruskal & Prim's Algorithms

💡 *Practice:*
Detect Cycle, Find Connected Components

⚙️ 7. Searching & Sorting

🔸 Searching

- Linear Search
- Binary Search
  💡 *Practice:* Search element in sorted array

🔸 Sorting

- Basic Sorting: Bubble, Selection, Insertion
- Advanced: Merge Sort, Quick Sort, Heap Sort
- Counting Sort, Radix Sort

💡 *Practice:* Sort array aur number of swaps count karna

🧩 8. Hashing

- Hash Table kya hota hai
- Hash functions aur collisions
- Collision handling (Chaining, Open Addressing)
- Applications of Hashing

💡 *Practice:* Two Sum Problem, Frequency Count, Subarray Sum = K

🚀 9. Advanced Algorithms

- **Dynamic Programming (DP)**
  - Top-down (Memoization)
  - Bottom-up (Tabulation)
  - Problems: Fibonacci, 0/1 Knapsack, LCS, Coin Change
- **Greedy Algorithms**
  - Activity Selection, Huffman Coding
- **Divide and Conquer**
  - Merge Sort, Quick Sort
- **Bit Manipulation** basics

💡 *Practice:* Minimum Coins, Power of Two

💻 10. Practice and Interview Preparation

- Topic-wise problem solving (LeetCode, GFG, CodeStudio, HackerRank)
- Mock interviews aur contests practice
- Daily 2–3 problems solve karne ka habit

🎯 11. Most Asked Interview Topics

- Arrays & Strings
- Linked List
- Recursion / Backtracking
- Binary Search
- Hashing (Maps, Sets)
- Trees / Graphs / DP