**DSA concepts and problems required for the fresher.  
Got it 👍. If you’re preparing DSA (Data Structures & Algorithms) as a fresher, interviewers will mainly test your problem-solving ability, clarity of concepts, and coding skills. Let me break it down into concepts + common problems you should prepare:**

**📌 1. Arrays & Strings**

**Concepts:**

* **Basics: Traversal, insertion, deletion, searching**
* **Prefix sum, sliding window**
* **Two pointers technique**
* **Sorting (basic + custom logic)**

**Problems:**

* **Reverse an array/string**
* **Find min & max in array**
* **Kadane’s algorithm (Maximum Subarray Sum)**
* **Move zeroes to end**
* **Two Sum problem**
* **Longest substring without repeating characters**
* **Anagram check**
* **Rotate array (left/right rotations)**

**📌 2. Linked List**

**Concepts:**

* **Singly & Doubly Linked Lists**
* **Traversal, insertion, deletion**
* **Fast & slow pointer technique**

**Problems:**

* **Reverse a linked list (iterative + recursive)**
* **Detect a cycle in linked list (Floyd’s algorithm)**
* **Find middle element**
* **Merge two sorted linked lists**
* **Remove Nth node from end**

**📌 3. Stack & Queue**

**Concepts:**

* **Implementation using arrays & linked lists**
* **Circular queue**
* **Applications (infix, postfix, prefix)**

**Problems:**

* **Balanced parentheses**
* **Next greater element**
* **Implement stack using queues (and vice versa)**
* **LRU Cache (advanced but common)**

**📌 4. Recursion & Backtracking**

**Concepts:**

* **Base case, recursion tree**
* **Subset generation, permutation generation**
* **Backtracking for constraint problems**

**Problems:**

* **Factorial, Fibonacci using recursion**
* **Tower of Hanoi**
* **N-Queens problem**
* **Rat in a Maze / Sudoku Solver**
* **Generate all subsets / permutations**

**📌 5. Searching & Sorting**

**Concepts:**

* **Binary search & variations**
* **Sorting algorithms (Bubble, Selection, Insertion, Merge, Quick, Counting, etc.)**

**Problems:**

* **Binary Search applications:**
  + **First & last occurrence of element**
  + **Square root using binary search**
  + **Search in rotated sorted array**
* **Merge intervals**
* **Kth largest/smallest element**

**📌 6. Hashing & Maps**

**Concepts:**

* **HashMap / Dictionary usage**
* **Frequency count, prefix sums with hash**

**Problems:**

* **Two Sum using HashMap**
* **Subarray sum = K**
* **Longest consecutive sequence**
* **Majority element (Moore’s Voting Algorithm)**

**📌 7. Trees & Binary Search Trees (BST)**

**Concepts:**

* **Traversals: Inorder, Preorder, Postorder, Level order**
* **BST properties**
* **Recursion in trees**

**Problems:**

* **Height of binary tree**
* **Lowest Common Ancestor (LCA)**
* **Check if binary tree is balanced**
* **Diameter of binary tree**
* **Validate BST**

**📌 8. Graphs**

**Concepts:**

* **BFS, DFS**
* **Representation (Adjacency list, matrix)**
* **Connected components**
* **Shortest path algorithms (Dijkstra, Bellman-Ford, Floyd-Warshall)**
* **Minimum Spanning Tree (Kruskal, Prim)**

**Problems:**

* **Number of islands**
* **Detect cycle in graph (directed & undirected)**
* **Topological sort**
* **Shortest path in unweighted graph (BFS)**

**📌 9. Dynamic Programming (DP)**

**Concepts:**

* **Memoization vs Tabulation**
* **Overlapping subproblems, optimal substructure**

**Problems:**

* **Fibonacci (DP version)**
* **Climbing Stairs**
* **Longest Common Subsequence (LCS)**
* **Coin Change**
* **Knapsack (0/1 Knapsack)**
* **Minimum Path Sum in grid**

**📌 10. Other Important Topics**

* **Bit Manipulation (check if power of 2, count set bits)**
* **Greedy Algorithms (Activity Selection, Huffman Coding basics)**
* **Heap / Priority Queue problems (Top K frequent elements, Heap sort)**

**✅ Tip for Freshers:**

* **Practice LeetCode Easy + Medium level problems.**
* **Start with arrays & strings, then go to linked lists & stacks, later cover trees, graphs, and DP.**
* **Focus on time complexity & space complexity (most interviewers ask).**

**Do you want me to create a 90-day roadmap with daily/weekly DSA problems so you can prepare step by step before interviews?**