**DSA IMP QUESTIONS**

**2️⃣ Arrays**

* Traversal, insertion, deletion
* ⭐ Reverse array in place
* ⭐ Find min & max
* ⭐ Rotate array (by k steps, both directions)
* ⭐ Kadane’s Algorithm (max subarray sum)
* ⭐ Prefix Sum & Difference Array
* ⭐ Two Pointer Technique (pair sum, 3-sum, container with most water)
* ⭐ Sliding Window Technique (max sum subarray of size k)
* Merge two sorted arrays
* Move all zeros to end
* Dutch National Flag Problem (Sort 0,1,2)
* Subarray with given sum (positive & mixed integers)

**3️⃣ Strings**

* Reverse string
* Palindrome check
* ⭐ Anagram check
* ⭐ Count frequency of characters
* ⭐ Longest Palindromic Substring
* ⭐ String matching algorithms (Naive, KMP, Rabin-Karp)
* ⭐ Longest Common Prefix
* ⭐ Subsequence vs Substring problems
* ⭐ Sliding window on strings (longest substring without repeating characters)
* Pattern searching

**4️⃣ Searching & Sorting**

* ⭐ Linear Search
* ⭐ Binary Search (and on answers)
* ⭐ Search in rotated sorted array
* ⭐ Merge Sort
* ⭐ Quick Sort
* ⭐ Heap Sort
* ⭐ Counting Sort / Radix Sort (for integers)
* Upper bound / lower bound logic

**5️⃣ Recursion & Backtracking**

* ⭐ Factorial, Fibonacci (recursion)
* ⭐ Tower of Hanoi
* ⭐ N-Queens Problem
* ⭐ Rat in a Maze
* ⭐ Word Search (matrix)
* ⭐ Generate all subsets / permutations
* ⭐ Sudoku Solver
* ⭐ Backtracking template

**6️⃣ Linked List**

* ⭐ Singly Linked List – insertion, deletion, traversal
* ⭐ Reverse a linked list (iterative & recursive)
* ⭐ Detect cycle in linked list (Floyd’s Algorithm)
* ⭐ Merge two sorted linked lists
* ⭐ Remove Nth node from end
* ⭐ Middle of linked list (fast/slow pointer)
* Doubly Linked List basics

**7️⃣ Stack & Queue**

**Stack:**

* ⭐ Implement stack using array/linked list
* ⭐ Balanced Parentheses
* ⭐ Next Greater Element
* ⭐ Min Stack
* ⭐ Stock Span Problem
* ⭐ Largest Rectangle in Histogram
* ⭐ Evaluate postfix/prefix expressions

**Queue:**

* ⭐ Implement queue using array/linked list
* ⭐ Circular Queue
* ⭐ Deque (double-ended queue)
* ⭐ Sliding Window Maximum
* ⭐ Rotten Oranges problem

**8️⃣ Hashing**

* ⭐ HashMap & HashSet basics
* ⭐ Frequency counting problems
* ⭐ Two Sum
* ⭐ Subarray with sum = K
* ⭐ Longest consecutive sequence
* ⭐ Detect duplicates
* Group Anagrams

**9️⃣ Trees**

* ⭐ Binary Tree traversal (inorder, preorder, postorder, level order)
* ⭐ Height of tree
* ⭐ Diameter of tree
* ⭐ Lowest Common Ancestor (LCA)
* ⭐ Binary Search Tree (insert, delete, search)
* ⭐ Validate BST
* ⭐ Top View / Bottom View of Tree
* ⭐ Serialize and Deserialize Binary Tree

**🔟 Heaps/Priority Queue**

* ⭐ Min Heap & Max Heap implementation
* ⭐ Heapify
* ⭐ Kth largest/smallest element
* ⭐ Heap Sort
* ⭐ Merge K sorted arrays
* ⭐ Top K frequent elements

**1️⃣1️⃣ Graphs**

* ⭐ BFS & DFS traversal
* ⭐ Detect cycle in graph (BFS/DFS)
* ⭐ Dijkstra’s Algorithm
* ⭐ Bellman-Ford Algorithm
* ⭐ Floyd-Warshall Algorithm
* ⭐ Kruskal’s & Prim’s MST
* ⭐ Topological Sort (Kahn’s Algorithm & DFS)
* ⭐ Connected components (Union-Find)

**1️⃣2️⃣ Dynamic Programming (DP)**

* ⭐ Fibonacci (top-down & bottom-up)
* ⭐ 0/1 Knapsack Problem
* ⭐ Coin Change
* ⭐ Longest Common Subsequence (LCS)
* ⭐ Longest Increasing Subsequence (LIS)
* ⭐ Matrix Chain Multiplication
* ⭐ Edit Distance
* ⭐ Minimum Path Sum
* ⭐ Partition Equal Subset Sum

**1️⃣3️⃣ Advanced Topics**

* Disjoint Set Union (Union-Find)
* Trie (Prefix Tree)
* Segment Tree (Range queries)
* Binary Indexed Tree (Fenwick Tree)
* Bit Manipulation (subset generation, XOR tricks)
* String hashing & Z-algorithm

**ADVANCED QUESTIONS**

**1️⃣ Basics / Mathematics**

* Time & Space Complexity Analysis (Amortized analysis too)
* GCD / LCM (Euclidean Algorithm)
* Sieve of Eratosthenes (Prime numbers)
* Modular exponentiation
* Bitwise operations (AND, OR, XOR, shifts)
* Count set bits (Brian Kernighan’s algo)
* Power of 2 check
* Fast exponentiation (binary exponentiation)
* Matrix exponentiation (for Fibonacci, paths)

**2️⃣ Arrays**

*(Core already listed before, extra below)*

* Merge intervals
* Find duplicates without extra space (Floyd’s cycle method)
* Majority element (Boyer–Moore Voting Algorithm)
* Minimum swaps to sort
* Trapping Rain Water (two-pointer & stack method)
* Product of array except self
* Spiral traversal of matrix
* Rotate matrix 90°
* Search in 2D matrix
* Maximum product subarray
* Longest subarray with sum divisible by k

**3️⃣ Strings**

*(Extra beyond core)*

* Count and say sequence
* Z-algorithm for pattern matching
* Manacher’s algorithm (Longest Palindromic Substring in O(n))
* Minimum window substring
* String compression (in-place)
* Wildcard pattern matching
* Regular expression matching (DP)

**4️⃣ Searching & Sorting**

* Ternary search
* Median of two sorted arrays
* Search in infinite sorted array
* Aggressive cows / Painter’s partition (Binary search on answer)

**5️⃣ Recursion & Backtracking**

* Combination Sum problems
* Word Break problem
* Letter combinations of a phone number
* Generate balanced parentheses
* Partition array to k subsets

**6️⃣ Linked List**

* Flatten a linked list
* Clone a linked list with random pointers
* Sort a linked list (merge sort)
* Add two numbers represented by linked lists

**7️⃣ Stack & Queue**

* Next smaller element
* Largest rectangle under histogram
* Min stack with O(1) space optimization
* Implement queue using stacks & stack using queues

**8️⃣ Hashing**

* Longest substring with at most K distinct characters
* Count subarrays with given XOR
* First non-repeating character in stream

**9️⃣ Trees**

*(Extra beyond core)*

* Invert/Flip a binary tree
* Path sum problems
* Morris Traversal (O(1) space traversal)
* Boundary traversal of binary tree
* Serialize/Deserialize tree

**🔟 Heaps**

* Median of a data stream
* Merge K sorted linked lists
* Sliding window median

**1️⃣1️⃣ Graphs**

*(Extra beyond core)*

* Bipartite graph check
* Bridges in graph (Tarjan’s Algorithm)
* Articulation points
* Kosaraju’s algorithm (SCC)
* Bellman-Ford negative cycle detection

**1️⃣2️⃣ Dynamic Programming**

*(Extra beyond core)*

* Egg Dropping problem
* Word break (DP approach)
* Burst Balloons problem
* Palindrome partitioning (min cuts)
* Count of Palindromic Substrings

**1️⃣3️⃣ Advanced Data Structures**

* Disjoint Set Union (Union-Find with path compression & union by rank)
* Trie with insert/search/prefix matching
* Segment Tree (Range min/max/sum queries)
* Lazy propagation in segment tree
* Binary Indexed Tree (Fenwick Tree)
* LRU Cache (LinkedHashMap / custom)
* Sparse Table (range queries without updates)

**1️⃣4️⃣ Special Patterns**

* Meet in the middle
* Prefix sum on 2D grid
* Monotonic stack & queue patterns
* Sliding window on arrays/strings
* Binary search on real numbers (precision problems)