**DSA Concepts:**

**Beginner Topics**

* Arrays & Strings
  + Sliding Window
  + Two Pointers
  + Prefix Sum
  + Kadane’s Algorithm (Max Subarray)
  + Rotate, Reverse, and Search (Binary Search on Rotated Arrays)
  + In-place Modifications
  + Subarrays & Substrings Patterns (Unique/Longest)
* Linked Lists
  + Reversal (Entire List / K-Groups)
  + Detect Cycle (Floyd’s Algorithm)
  + Merge Two Sorted Lists
  + Intersection Point of Two Lists
  + Remove N-th node from End
  + Palindrome Linked List

**Intermediate Topics**

* Stacks & Queues
  + Valid Parantheses / Balancing Symbols
  + Monotonic Stack (Next Greater Element)
  + Design stack with getMin()
  + Sliding Window Maximum (Deque)
  + Implement Queue using Stacks (and vice versa)
  + LRU Cache (Design Problems)
* Hashing & Hash Maps/Sets
  + Two Sum, Three Sum, Four Sum
  + Group Anagrams
  + Subarrays with Sum K (Prefix Hashing)
  + Longest Consecutive Sequence
  + Design Data Structures (LRU, LFU Cache)
* Trees & Binary Trees
  + Inorder, Preorder, Postorder Traversal (Iterative/Recursive)
  + Level Order Traversal (BFS)
  + Depth-first Search (DFS)
  + Diameter of Binary Tree
  + Lowest Common Ancestor (LCA)
  + Symmetric & Balanced Trees
  + Serialize and Deserialize Binary Tree
* Binary Search Trees (BST)
  + Search, Insert, Delete Node
  + Validate BST
  + Kth Smallest / Largest in BST
  + BST to Doubly Linked List
  + Range Sum of BST
* Heaps / Priority Queues
  + Kth Largest Element in Array / Stream
  + Merge K Sorted Lists / Arrays
  + Top K Frequent Elements
  + Min / Max Heaps Usage Patterns
* Recursion & Backtracking
  + Subsets / Permutations / Combinations
  + N-Queens Problem
  + Sudoku Solver
  + Word Search
  + Combination Sum Variations
* Divide & Conquer
  + Merge Sort / Quick Sort
  + Search in Rotated Sorted Array
  + Median of Two Sorted Arrays (Hard)
  + Count Inversions in Array

**Advanced Topics**

* Dynamic Programming (DP)
  + 0/1 Knapsack, Unbounded Knapsack
  + Longest Common Subsequence (LCS)
  + Longest Increasing Subsequence (LIS)
  + Matrix DP (Unique Paths, Coin Change, etc.)
  + DP on Strings (Edit Distance, Palindromic Substrings)
  + DP on Subsequence (Subset Sum, Partition Equal Subset Sum)
  + DP with States (House Robber, Paint House)

* Greedy Algorithms
  + Activity Selection Problem
  + Fractional Knapsack
  + Huffman Encoding
  + Interval Scheduling (Merge Intervals, Non-overlapping Intervals)
* Graph Algorithms
  + BFS & DFS (Matric and Graph Traversal)
  + Dijkstra’s Algorithm
  + Topological Sort (Kahn’s Algorithm, DFS-based)
  + Union-Find / Disjoint Set Union (DSU)
  + Kruskal’s & Prim’s MST
  + Detect Cycle in Graph (Directed / Undirected)
  + Number of Islands / Connected Components
* Tries (Prefix Trees)
  + Insert and Search Words
  + Auto-complete Suggestions
  + Word Search II
  + Longest Word in All Prefixes
* Bit Manipulation
  + Count Set Bits
  + Single Number (appears once, others twice)
  + XOR Patterns
  + Subsets using Bitmasks
* Sliding Window Advanced & Two Pointers Advanced
  + Longest Substring Without Repeating Characters
  + Minimum Window Substring
  + Longest Subarray Sum K (using HashMap + Prefix Sum)
* Advanced Topics (for Google / Meta)
  + Segment Trees / Fenwick Tree (Binary Indexed Tree)
  + Line Sweep Algorithms
  + Range Queries (Sparse Table)
  + Advanced DP (Bitmask DP, DP on Trees)
  + Heavy-Light Decomposition (HLD)

Array Methods and Properties:

Array Properties

|  |  |
| --- | --- |
| Property | Description |
| length | Returns the number of elements in the array |
| constructor | Returns the function that created the Array prototype |
| prototype | Allows adding properties/methods to Array objects |

Mutating (Modifies Original Array) Methods

|  |  |
| --- | --- |
| Method | Description |
| push(element) | Adds element(s) to the end |
| pop() | Removes and returns the last element |
| shift() | Removes and returns the first element |
| unshift(element) | Adds element(s) to the start |
| splice(start, deletecount, item1, …) | Add/Remove/Replace elements |
| reverse() | Reverses the array in-place |
| sort(compareFunction) | Sorts the array in-place |
| fill(value, start, end) | Fills elements with static value |
| copyWithin(target, start, end) | Copies sequence within array |

Non-Mutating (Does NOT Modify Array) Methods

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |