**Master Data Structures and Algorithms**

**About us**

We are a group of IITians who recently cracked companies like Flipkart, Jaguar Land Rover, Amazon, Vedantu, JioSaavn, Zomato, Arista etc, and are guiding students.

You can join our telegram group to get in touch and ask your queries: [**https://t.me/dream\_placement**](https://t.me/dream_placement)

**Youtube Channel:** [**https://www.youtube.com/c/KayDee24/featured**](https://www.youtube.com/c/KayDee24/featured)

The DS-Algo questions list is prepared by us after analyzing all the websites available and taking insights from our own preparation and experiences. This question set if done properly is enough to prepare for maximum companies.

**Pre-requisites:**

* 1 Coding language (C++, Java or Python) along with basic functionalities like array declaration, loops, if-else statements, and declaring variables.
* Sorting, Hashing (Basic)
* Basic time and space complexities of loop traversals & sorting.
* For C++ users: STL like Vector, Map, Set (to implement hashing), Sort function (to sort).

Note: Vector is nothing but an array just with some flexibilities like inserting as many elements without initially declaring size. Going through geeksforgeeks STL will be enough for getting comfortable with there implementation.

Refer Tushar Roy, Abdul Bari and Jenny’s Lectures videos in case you need help for most of your doubts.

1. **ARRAYS:**
2. [Rotate an array](https://practice.geeksforgeeks.org/problems/rotate-array-by-n-elements/0)
3. [Leaders in an array](https://practice.geeksforgeeks.org/problems/leaders-in-an-array-1587115620/1)
4. [Pascal triangle](https://leetcode.com/problems/pascals-triangle/)
5. [Max Sum Subarray](https://www.interviewbit.com/problems/max-sum-contiguous-subarray/)
6. [Find first missing and repeating number](https://practice.geeksforgeeks.org/problems/find-missing-and-repeating2512/1)
7. [Stock Buy and Sell](https://practice.geeksforgeeks.org/problems/stock-buy-and-sell-1587115621/1)
8. [Majority Element](https://leetcode.com/problems/majority-element/) (Study all possible methods)
9. [Minimum Platforms](https://practice.geeksforgeeks.org/problems/minimum-platforms-1587115620/1)
10. [Merge overlapping intervals](https://leetcode.com/problems/merge-intervals/)
11. [Elements on left smaller and on right greater](https://practice.geeksforgeeks.org/problems/unsorted-array4925/1)
12. [Trapping Rainwater](https://practice.geeksforgeeks.org/problems/trapping-rain-water-1587115621/1)
13. [Maximum Index](https://practice.geeksforgeeks.org/problems/maximum-index-1587115620/1)
14. [Move all zeros to the end of the array](https://practice.geeksforgeeks.org/problems/move-all-zeroes-to-end-of-array0751/1) (Also try: [Remove duplicates from the sorted array](https://practice.geeksforgeeks.org/problems/remove-duplicate-elements-from-sorted-array/1))
15. [Max sum subarray of size k](https://practice.geeksforgeeks.org/problems/max-sum-subarray-of-size-k5313/1)
16. [Chocolate Distribution Problem](https://practice.geeksforgeeks.org/problems/chocolate-distribution-problem/0)
17. [Find first element occurring k times](https://www.geeksforgeeks.org/first-element-occurring-k-times-array/)
18. [Minimum swaps to sort](https://practice.geeksforgeeks.org/problems/minimum-swaps/1)
19. [Sort elements of an array by frequency](https://practice.geeksforgeeks.org/problems/sorting-elements-of-an-array-by-frequency/0)
20. [Product of an array except for self](https://leetcode.com/problems/product-of-array-except-self/)
21. [Rearrange array alternatively](https://practice.geeksforgeeks.org/problems/-rearrange-array-alternately-1587115620/1)
22. [Find repeating element in an array](https://www.geeksforgeeks.org/find-duplicates-in-on-time-and-constant-extra-space/) (Also try: [Smallest positive missing number](https://practice.geeksforgeeks.org/problems/smallest-positive-missing-number-1587115621/1))

**2D Array:**

1. [Spiral matrix](https://practice.geeksforgeeks.org/problems/spirally-traversing-a-matrix-1587115621/1)
2. [Rotate matrix](https://practice.geeksforgeeks.org/problems/rotate-by-90-degree-1587115621/1)
3. [Set matrix zeroes](https://www.interviewbit.com/problems/set-matrix-zeros/)
4. [Max-sum square sub-matrix](https://www.interviewbit.com/problems/maximum-sum-square-submatrix/)

**Advance:**

-[Minimize the heights](https://practice.geeksforgeeks.org/problems/minimize-the-heights3351/1)

-[3 partition](https://www.geeksforgeeks.org/split-array-three-equal-sum-subarrays/#:~:text=Determine%20if%20array%20A%20can,%5D)%2C%20otherwise%20print%20%2D1.)

-[Maximum unsorted subarray](https://practice.geeksforgeeks.org/problems/length-unsorted-subarray3022/1)

-[Next Permutation](https://practice.geeksforgeeks.org/problems/next-permutation5226/1)

[-Kth smallest element](https://practice.geeksforgeeks.org/problems/kth-smallest-element5635/1)

-[Max Sum Triplet](https://www.interviewbit.com/problems/maximum-sum-triplet/)

**2. STRINGS:**

Note: Strings are very similar to arrays because a string is nothing but an array of characters only.

1. [Palindrome string](https://practice.geeksforgeeks.org/problems/palindrome-string0817/1)
2. [Longest Common Prefix](https://leetcode.com/problems/longest-common-prefix/https://leetcode.com/problems/longest-common-prefix/)
3. [Zig zag string](https://practice.geeksforgeeks.org/problems/concatenation-of-zig-zag-string-in-n-rows/0)
4. [Anagrams](https://practice.geeksforgeeks.org/problems/anagram-1587115620/1)
5. [Atoi](https://practice.geeksforgeeks.org/problems/implement-atoi/1)
6. [Number following a pattern](https://practice.geeksforgeeks.org/problems/number-following-a-pattern/0)
7. [Count and say](https://www.interviewbit.com/problems/count-and-say/)
8. [Find one extra character in a string](https://www.geeksforgeeks.org/find-one-extra-character-string/)
9. [Find if strings are rotations of each other or not](https://practice.geeksforgeeks.org/problems/check-if-strings-are-rotations-of-each-other-or-not/0)

**Advance:**

- [Longest Substring without repeating character](https://practice.geeksforgeeks.org/problems/length-of-the-longest-substring3036/1)

-[Smallest Window containing all character of another string](https://practice.geeksforgeeks.org/problems/smallest-window-in-a-string-containing-all-the-characters-of-another-string-1587115621/1)

-[Design a tiny URL or URL shortener](https://practice.geeksforgeeks.org/problems/design-a-tiny-url-or-url-shortener/0)

-[Longest Palindrome string](https://practice.geeksforgeeks.org/problems/longest-palindrome-in-a-string/0) (Do it in O(1) space and O(n^2) time)

**3. Hashing and 2 Pointer**

(Famous methods to solve questions in efficient space and time complexities)**:**

**Hashing:**

1. [2 out of 3](https://www.interviewbit.com/problems/two-out-of-three/#:~:text=Given%20are%20Three%20arrays%20A,out%20of%20the%203%20arrays.&text=A%2C%20B%2C%20C%20may%20or,not%20have%20pairwise%20distinct%20elements.)
2. [Valid Sudoko](https://practice.geeksforgeeks.org/problems/is-sudoku-valid4820/1/)
3. [2 sum](https://practice.geeksforgeeks.org/problems/key-pair5616/1), [3 sum](https://practice.geeksforgeeks.org/problems/triplet-sum-in-array-1587115621/1), [4 sum](https://practice.geeksforgeeks.org/problems/find-all-four-sum-numbers1732/1)
4. [Largest subarray with given sum k](https://practice.geeksforgeeks.org/problems/longest-sub-array-with-sum-k0809/1)
5. [Largest subarray with equal 0s and 1s](https://practice.geeksforgeeks.org/problems/largest-subarray-of-0s-and-1s/1)
6. [Distinct elements in every window of size k](https://practice.geeksforgeeks.org/problems/count-distinct-elements-in-every-window/1)
7. [Subarray with given xor](https://www.interviewbit.com/problems/subarray-with-given-xor/)

**2 Pointer:**

1. [Container with most water](https://leetcode.com/problems/container-with-most-water/)
2. [Remove duplicates from a sorted array](https://practice.geeksforgeeks.org/problems/remove-duplicate-elements-from-sorted-array/1)
3. [Two numbers with sum closest to zero](https://practice.geeksforgeeks.org/problems/two-numbers-with-sum-closest-to-zero1737/1)
4. [3 elements sum lies closest to a given k](https://www.geeksforgeeks.org/find-a-triplet-in-an-array-whose-sum-is-closest-to-a-given-number/)
5. [No of triangles formed from an array](https://www.geeksforgeeks.org/find-number-of-triangles-possible/)
6. [3 sorted arrays](https://www.interviewbit.com/problems/array-3-pointers/)
7. [Geek collect the balls](https://practice.geeksforgeeks.org/problems/geek-collects-the-balls5515/1)

**Advance:**

-[Points in a straight line](https://www.interviewbit.com/problems/points-on-the-straight-line/)

-[Subarray with B odd numbers](https://www.interviewbit.com/problems/subarray-with-b-odd-numbers/)

-[Pair with given diff](https://practice.geeksforgeeks.org/problems/find-pair-given-difference/0)

-[Max ones after Modification](https://practice.geeksforgeeks.org/problems/find-pair-given-difference/0)

-[Smallest distinct window](https://practice.geeksforgeeks.org/problems/smallest-distant-window3132/1)

**4. Linked List:**

1. [Reverse a Linked List](https://practice.geeksforgeeks.org/problems/reverse-a-linked-list/1)
2. [Reverse a Linked List in groups](https://practice.geeksforgeeks.org/problems/reverse-a-linked-list-in-groups-of-given-size/1)
3. [Reverse alternate k nodes](https://www.interviewbit.com/problems/reverse-alternate-k-nodes/)
4. [Palindrome Linked List](https://practice.geeksforgeeks.org/problems/check-if-linked-list-is-pallindrome/1)
5. [Remove duplicates from Linked List](https://www.geeksforgeeks.org/remove-duplicates-from-an-unsorted-linked-list/)
6. [Print mid of Linked List](https://practice.geeksforgeeks.org/problems/finding-middle-element-in-a-linked-list/1)
7. [Find nth node from end of linked list](https://practice.geeksforgeeks.org/problems/nth-node-from-end-of-linked-list/1)
8. [Remove loop in linked list and find that point](https://practice.geeksforgeeks.org/problems/nth-node-from-end-of-linked-list/1)
9. [Delete node without head pointer](https://practice.geeksforgeeks.org/problems/delete-without-head-pointer/1)
10. [Sort list](https://practice.geeksforgeeks.org/problems/sort-a-linked-list/1)
11. [Merge k sorted Linked List](https://practice.geeksforgeeks.org/problems/sort-a-linked-list/1)
12. [Intersection point in Y shaped Linked List](https://practice.geeksforgeeks.org/problems/intersection-point-in-y-shapped-linked-lists/1)
13. [Clone a Linked List](https://practice.geeksforgeeks.org/problems/clone-a-linked-list-with-next-and-random-pointer/1)

**5. Stacks and Queues:**

**Stacks:**

1. [Implement stack using array](https://practice.geeksforgeeks.org/problems/implement-stack-using-array/1#:~:text=You%20are%20required%20to%20complete,from%20the%20pop()%20method.)
2. [Reverse a sentence using stack](https://practice.geeksforgeeks.org/problems/reverse-a-string-using-stack/1)
3. [Balanced parentheses](https://practice.geeksforgeeks.org/problems/parenthesis-checker2744/1)
4. [Longest balanced parenthesis](https://practice.geeksforgeeks.org/problems/longest-valid-parentheses5657/1/?page=2&sortBy=submissions&category%5B%5D=Stack&query=page2sortBysubmissionscategory%5B%5DStack)
5. [Rain Water Trapping](https://practice.geeksforgeeks.org/problems/trapping-rain-water-1587115621/1)
6. [Stock span problem](https://practice.geeksforgeeks.org/problems/stock-span-problem-1587115621/1)
7. [Next larger element](https://practice.geeksforgeeks.org/problems/next-larger-element-1587115620/1) (Also try next smaller element)
8. [Max rectangle area in histogram](https://practice.geeksforgeeks.org/problems/maximum-rectangular-area-in-a-histogram-1587115620/1) (Hint: Using the concept of next larger element)
9. [Minimum element of stack](https://practice.geeksforgeeks.org/problems/special-stack/1)
10. [Celebrity problem](https://practice.geeksforgeeks.org/problems/the-celebrity-problem/1)
11. [Evaluation of expressions](https://www.geeksforgeeks.org/expression-evaluation/)

**Queue:**

1. [Queue using 2 stacks](https://practice.geeksforgeeks.org/problems/queue-using-two-stacks/1)
2. [Decode string](https://leetcode.com/problems/decode-string/)
3. [Reverse First K elements of Queue](https://practice.geeksforgeeks.org/problems/reverse-first-k-elements-of-queue/1)
4. [First non-repeating character in a stream](https://practice.geeksforgeeks.org/problems/first-non-repeating-character-in-a-stream1216/1)
5. [Rotten oranges](https://practice.geeksforgeeks.org/problems/rotten-oranges2536/1)
6. [Max of all subarrays of size k](https://practice.geeksforgeeks.org/problems/maximum-of-all-subarrays-of-size-k3101/1) (Deque)
7. [LRU cache](https://practice.geeksforgeeks.org/problems/lru-cache/1)

**Heap** (Actually a tree but can be implemented using STL Priority Queue)**:**

1. [Merge k sorted arrays](https://www.interviewbit.com/problems/merge-k-sorted-lists/)
2. [Min Cost Ropes](https://practice.geeksforgeeks.org/problems/minimum-cost-of-ropes-1587115620/1)
3. [Distinct numbers in a window](https://www.interviewbit.com/problems/distinct-numbers-in-window/)
4. [N max pair combination](https://www.interviewbit.com/problems/n-max-pair-combinations/)

**6. Recursion & Backtracking**

(Focus only on the basics, no need to go very advance)

**Recursion:**

1. [Print 1 to n without loop](https://practice.geeksforgeeks.org/problems/print-1-to-n-without-using-loops-1587115620/1/?difficulty%5B%5D=-2&page=1&category%5B%5D=Recursion&query=difficulty%5B%5D-2page1category%5B%5DRecursion)
2. [Print factorial of a number](https://practice.geeksforgeeks.org/problems/factorial5739/1)
3. [Nth fibonacci number](https://practice.geeksforgeeks.org/problems/nth-fibonacci-number1335/1)
4. [Print all possible subsequences](https://practice.geeksforgeeks.org/problems/print-all-possible-strings/1/?page=1&category%5B%5D=Recursion&query=page1category%5B%5DRecursion) ([Best video solution](https://www.youtube.com/watch?v=KCEPvdLqlYI))
5. [Print all strings](https://practice.geeksforgeeks.org/problems/print-all-possible-strings/1/?page=1&category%5B%5D=Recursion&query=page1category%5B%5DRecursion)
6. [Tower of Hanoi](https://practice.geeksforgeeks.org/problems/tower-of-hanoi-1587115621/1/?difficulty%5B%5D=1&page=1&sortBy=submissions&category%5B%5D=Recursion&query=difficulty%5B%5D1page1sortBysubmissionscategory%5B%5DRecursion)

**Backtracking:**

Beginners first watch [t](https://www.youtube.com/watch?v=S3rnLLHl0PM)his: <https://www.youtube.com/watch?v=S3rnLLHl0PM>

1. [Subset](https://www.interviewbit.com/problems/subset/)
2. [Subset 2](https://www.interviewbit.com/problems/subsets-ii/)

**8. Binary Search**

(There are 2 ways to implement this algorithm, either by recursion or loop, you may use either)

1. [Square root](https://www.interviewbit.com/problems/square-root-of-integer/)
2. [Rotated sorted array search](https://www.interviewbit.com/problems/rotated-sorted-array-search/)
3. [Find element in a bitonic array](https://leetcode.com/problems/peak-index-in-a-mountain-array/)
4. [Left most and right most index](https://practice.geeksforgeeks.org/problems/find-first-and-last-occurrence-of-x0849/1/?page=1&sortBy=submissions&category%5B%5D=Binary%20Search&query=page1sortBysubmissionscategory%5B%5DBinary%20Search)
5. [Matrix Search](https://www.interviewbit.com/problems/matrix-search/)
6. [Sorted matrix median](https://www.interviewbit.com/problems/matrix-median/)
7. [Painter Partition Problem](https://www.interviewbit.com/problems/painters-partition-problem/) / [Allocate Books](https://www.interviewbit.com/problems/allocate-books/)
8. [Usage of Upper](https://www.geeksforgeeks.org/upper_bound-in-cpp/)/[Lower bound STL](https://www.geeksforgeeks.org/lower_bound-in-cpp/)

**9. Trees**

(We will deal with binary tree only for almost all the questions at our level)

1. Create a [Binary Tree](https://www.geeksforgeeks.org/binary-tree-set-1-introduction/) and [Binary Search Tree (BST)](https://www.geeksforgeeks.org/binary-search-tree-set-1-search-and-insertion/)
2. [Search in a BST](https://leetcode.com/problems/search-in-a-binary-search-tree/)
3. [Find height of a tree](https://practice.geeksforgeeks.org/problems/height-of-binary-tree/1)/ [Min Depth](https://practice.geeksforgeeks.org/problems/minimum-depth-of-a-binary-tree/1)
4. [Level Order Traversal](https://practice.geeksforgeeks.org/problems/level-order-traversal/1) / [Zig-Zag Level Order Traversal](https://practice.geeksforgeeks.org/problems/zigzag-tree-traversal/1)
5. [Pre](https://practice.geeksforgeeks.org/problems/preorder-traversal/1)/ [In](https://practice.geeksforgeeks.org/problems/inorder-traversal/1) / [Post](https://practice.geeksforgeeks.org/problems/postorder-traversal/1) Order Traversal
6. Iterative [Pre](https://www.geeksforgeeks.org/iterative-preorder-traversal/)/ [In](https://www.geeksforgeeks.org/inorder-tree-traversal-without-recursion/) / [Post](https://www.geeksforgeeks.org/iterative-postorder-traversal/) order traversal
7. Given 2 traversals, create binary tree [I](https://www.geeksforgeeks.org/iterative-postorder-traversal/), [II](https://practice.geeksforgeeks.org/problems/construct-tree-1/1) , [III](https://practice.geeksforgeeks.org/problems/tree-from-postorder-and-inorder/1)
8. [Check if 2 trees are identical](https://practice.geeksforgeeks.org/problems/determine-if-two-trees-are-identical/1)
9. [Count number of leaves of a tree](https://practice.geeksforgeeks.org/problems/count-leaves-in-binary-tree/1) / [Sum of a tree](https://practice.geeksforgeeks.org/problems/sum-of-binary-tree/1)
10. [Left](https://practice.geeksforgeeks.org/problems/left-view-of-binary-tree/1) / [Right View of a tree](https://practice.geeksforgeeks.org/problems/right-view-of-binary-tree/1)
11. [Mirror Tree](https://practice.geeksforgeeks.org/problems/mirror-tree/1)
12. [Check for balanced tree](https://practice.geeksforgeeks.org/problems/check-for-balanced-tree/1)
13. [Symmetric Tree](https://practice.geeksforgeeks.org/problems/symmetric-tree/1)
14. [Diameter of a tree](https://practice.geeksforgeeks.org/problems/diameter-of-binary-tree/1)
15. [Lowest Common Ancestor (LCA)](https://practice.geeksforgeeks.org/problems/lowest-common-ancestor-in-a-binary-tree/1)
16. [All nodes at K distance from root](https://practice.geeksforgeeks.org/problems/nodes-at-given-distance-in-binary-tree/1)
17. [Print path from root to a particular node](https://www.interviewbit.com/problems/path-to-given-node/)
18. [Vertical order traversal](https://www.interviewbit.com/problems/vertical-order-traversal-of-binary-tree/)
19. [Top](https://practice.geeksforgeeks.org/problems/top-view-of-binary-tree/1) / [Bottom view](https://practice.geeksforgeeks.org/problems/bottom-view-of-binary-tree/1)
20. [Boundary Traversal](https://practice.geeksforgeeks.org/problems/boundary-traversal-of-binary-tree/1)
21. [Sum of all left leaf nodes](https://www.geeksforgeeks.org/find-sum-left-leaves-given-binary-tree/)
22. [Swap leaf nodes](https://practice.geeksforgeeks.org/problems/exchange-the-leaf-nodes/1)
23. [Max path sum between 2 leaves](https://practice.geeksforgeeks.org/problems/maximum-path-sum/1)
24. [Remove all the half nodes](https://practice.geeksforgeeks.org/problems/remove-half-nodes/1)

**10. Graphs**

(BFS and DFS are the 2 main pillars of most of the graph questions)

Understand ways of representing graphs first.

1. [Breadth First Search (BFS)](https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/)
2. [Depth First Search (DFS)](https://www.geeksforgeeks.org/depth-first-search-or-dfs-for-a-graph/)
3. [Find no. of islands](https://leetcode.com/problems/number-of-islands/)
4. [Maximum area island](https://leetcode.com/problems/max-area-of-island/)
5. [Snake and Ladder](https://www.interviewbit.com/problems/snake-ladder-problem/)
6. [Word Ladder I](https://www.interviewbit.com/problems/word-ladder-i/) and [II](https://www.interviewbit.com/problems/word-ladder-ii/)
7. [Shortest path in a matrix](https://leetcode.com/problems/shortest-path-in-binary-matrix/)
8. [Detect Cycle in a directed graph](https://practice.geeksforgeeks.org/problems/detect-cycle-in-a-directed-graph/1)
9. [Detect Cycle in an undirected graph](https://practice.geeksforgeeks.org/problems/detect-cycle-in-an-undirected-graph/1)
10. [Topological Sort](https://practice.geeksforgeeks.org/problems/topological-sort/1)
11. Shortest path from 1 source to all other sources or any particular source in weighted graph with +Ve Edges - [Dijkstra algo](https://practice.geeksforgeeks.org/problems/shortest-path-from-1-to-n0156/1) | -Ve Edges included - [Bellman ford algo](https://practice.geeksforgeeks.org/problems/negative-weight-cycle3504/1)

**Advance:**

1. [Minimum Spanning Tree](https://www.interviewbit.com/problems/commutable-islands/)
2. [Strongly connected components](https://www.geeksforgeeks.org/strongly-connected-components/)
3. [Articulation points](https://www.geeksforgeeks.org/articulation-points-or-cut-vertices-in-a-graph/)

**11. Dynamic Programming**

(Just a concept to solve questions efficiently, nothing fancy)

1. [Count ways to reach nth stair](https://practice.geeksforgeeks.org/problems/count-ways-to-reach-the-nth-stair-1587115620/1)
2. [Thief Problem](https://practice.geeksforgeeks.org/problems/max-sum-without-adjacents2430/1)
3. [Longest Increasing Subsequence](https://practice.geeksforgeeks.org/problems/longest-increasing-subsequence-1587115620/1)
4. [Longest Common Subsequence](https://practice.geeksforgeeks.org/problems/longest-common-subsequence-1587115620/1)
5. [Minimum jumps array](https://practice.geeksforgeeks.org/problems/minimum-number-of-jumps-1587115620/1)
6. [Largest square formed](https://practice.geeksforgeeks.org/problems/largest-square-formed-in-a-matrix0806/1)
7. [0-1 Knapsack](https://practice.geeksforgeeks.org/problems/0-1-knapsack-problem0945/1)
8. [Coin Change](https://practice.geeksforgeeks.org/problems/coin-change2448/1)
9. [Coin Change II](https://practice.geeksforgeeks.org/problems/number-of-coins1824/1)
10. [Rod Cutting](https://practice.geeksforgeeks.org/problems/cutted-segments1642/1)
11. [Edit Distance](https://www.geeksforgeeks.org/edit-distance-dp-5/)
12. [Subset Sum](https://practice.geeksforgeeks.org/problems/subset-sum-problem2014/1)
13. [Longest Palindromic subsequence](https://leetcode.com/problems/longest-palindromic-subsequence/)
14. [Best time to buy and sell stock](https://practice.geeksforgeeks.org/problems/maximum-profit4657/1/)

**Advance:**

-[Word Break](https://practice.geeksforgeeks.org/problems/word-break1352/1)

-[Min sum partition](https://practice.geeksforgeeks.org/problems/minimum-sum-partition3317/1)

Summary:

– Mostly Max or Min Nikalana hoga

1. Think of a recursion solution
2. Create an array or a matrix and try to fit in the question
3. Fill the array/matrix by feeling the problem ;p

**10. Bit manipulation**

1. [Trailing Zeroes](https://www.interviewbit.com/problems/trailing-zeroes/)
2. [Reverse bits](https://www.interviewbit.com/problems/reverse-bits/)
3. [Number of 1 bits](https://www.interviewbit.com/problems/number-of-1-bits/)
4. [Different bits sum pairwise](https://www.interviewbit.com/problems/different-bits-sum-pairwise/)
5. [XORing the subarrays](https://www.interviewbit.com/problems/xor-ing-the-subarrays/)

**Miscellaneous:**

1. [KMP algorithm](https://practice.geeksforgeeks.org/problems/longest-prefix-suffix2527/1)
2. [Merge 2 array without extra space](https://practice.geeksforgeeks.org/problems/merge-two-sorted-arrays-1587115620/1)
3. [Inversion count](https://practice.geeksforgeeks.org/problems/inversion-of-array-1587115620/1)
4. [Median of row wise sorted matrix](https://practice.geeksforgeeks.org/problems/median-in-a-row-wise-sorted-matrix1527/1)
5. [Median of an infinite stream](https://practice.geeksforgeeks.org/problems/find-median-in-a-stream-1587115620/1) & [Median of 2 sorted sets](https://practice.geeksforgeeks.org/problems/find-the-median0527/1)
6. [Rotate a matrix by 90 degrees](https://practice.geeksforgeeks.org/problems/rotate-by-90-degree-1587115621/1)
7. Tries Implementation

-------------------------------------------------------------------------------------------------------------------------------

**Problem Solving Approach:**

1. Read the question carefully and understand some test cases first.
2. Then try to think of solutions, try various approaches and test your approaches on test cases. Give at least 30 minutes to a question before opening the solution on paper.
3. Writing the code and submitting should be the last step once you are confident your approach is working on many test cases.
4. Patience is the key here and people who will struggle with the questions will learn much more as compared to ones who see the solution within a few minutes after reading the question.
5. The goal is to develop problem-solving skills through these questions and not mugging them up, so try build that instinct of learning something new from every question.
6. Your notes you make are your Bhagavad gita, keep revising them so that you don’t forget after a few months.