Ganesh Prasad

109 Followers About Follow

Interview Preparation Plan for 30 Days: 200+ coding and Behavioural questions

Ganesh Prasad Jul 12 · 12 min read ★

I have curated these questions into a list after reading interview experiences, geeksforgeeks, leetcode and several other sources… These questions cover data structures and algorithms and are frequent in coding interviews.

7 Questions a day, I think is achievable, As there are articles to read and then practise. Read and Practise.

Table of contents

- Day 4: Sliding Window Patten
- Day 5: Maths
- Day 6 & 7: Linked list
- Day 8: 2 Pointer
- Day 9: Fast & Slow Pointers
- Day 10: Stack
- Day 11: Queue
- Day 12: Hashing
- Day 13 & 14: Heap
- Day 15: Greedy
- Day 16: Recursion
- Day 17: Backtracking
- Day 18: Divide & Conquer
- Day 19: String
- Day 20 & 21: Binary Tree
- Day 22: Binary Search Tree
- Day 23: Mixed Topics
- Day 24: Graph
- Day 25 & 26: Dynamic Programming
- Day 27: Design Patterns
- Bits Manipulation (optional)
- Day 28 & 29: Theory (OS, CN, DB) **(To be added)**
- Day 30: Behavioural Interview **(To be added)**
- Day 30: Projects (How to discuss projects in interviews) **(To be added)**

Important Note:

- For each task/problem, I have written an article in simple language to make the pattern of the problem clear.
- I will recommend that you try the problems yourself after reading the article.
- **Important**: Try to get the problem pattern through the articles, as this will prepare you to tackle problems that follow similar patterns.
- These articles are in the order, Brute force solution > Efficient solution. In interviews, it is **crucial** to start with brute force and improve because **the**

interviewer wants to see your problem-solving skills. (**never** jump directly to the most efficient solution unless asked for)

- Bookmark this page for later use.
- Work on your typing speed daily (Go for at least 50 Words per minute).

> *According to me: The purpose of this article is not to make you able to solve* ***these questions*** *in your interview But to prepare you to tackle any problem.* ***(Remember ; understanding the patterns is the ultimate goal).***

For the first 27 days, we will do these coding questions, and in the remaining 3 days, we will cover behavioural interviews and theory.

(Go to Top)

Array (2 Easy — 3 Medium — 1 Hard)

The array is considered the most important topic in technical interviews; there are big companies that ask the majority of their array-based questions.

Problems:

Day 1.1: Sort an array of 0s, 1s and 2s — Easy

Given an array of 0s, 1s and 2s, sort it in ascending order. Read here .

Day 1.2: Find missing and repeating numbers — Easy

Given an array of size n with elements in the range [1, n], find the missing and repeating numbers if one element is repeated once. Read here .

- One more Simple XOR-based problem that might clear the concept of this operator. — **Single Number**.

Day 1.3: Product of array except self — Medium

Given an array, find another array index ***i*** is the product of all the elements of the array except element at index ***i***. read here .

Day 1.4: Setting zeros in a matrix — Medium

Given a 2D array, set the whole row and column to 0 if that row or column contains 0. In other words, if cell [i, j] is 0, then set the whole row ***i*** and col ***j*** to 0. read here .

Day 1.5: Merge two sorted arrays without using Extra space — Medium

Two sorted arrays are given, and you have to merge them into one sorted array without using extra space. #TODO .

Day 1.6: Maximum Circular Subarray Sum — Hard

Given an array with N integers in a circular manner, Find the maximum continuous subarray sum. #TODO

End of Day 1

(Go to Top)

Sorting (Arrays)

Problems:

Day 2.1: Majority element in an Array. — Easy

Day 2.2: Chocolate distribution problem. — Easy

Day 2.3: Find the Triplets with sum within the given range. — Medium

Day 2.4: Minimum number of Platforms — Medium

Day 2.5: Maximum index difference. — Medium

End of Day 2

(Go to Top)

Searching

Problems:

Day 3.1: Binary search in the forest for collecting woods. — Medium

Day 3.2: Find Missing Elements in the Second array that are present in the first array. — Medium

Day 3.3: Find the square root of x. — Medium

Day 3.4: Find Median of Two Sorted Array of Different Sizes. — Hard

End of Day 3

(Go to Top)

Matrix

Problems:

Day 3.5: Search in a Matrix. — Easy

Day 3.6: Sort a 2d Matrix Diagonally — Medium

Day 3.7: Traverse a Matrix in Spiral form. — Medium

Day 3.8: Rotate a Matrix by 90 degrees — Medium

End of Day 3

(Go to Top)

Sliding window pattern — (2 Easy, 2 Medium & 2 Hard)

Day 4.1: Maximum Sum Subarray of size K. — Easy

Day 4.2: Smallest Subarray with a given sum — Easy

Day 4.3: Longest Repeating String after K Replacement. — Medium

Day 4.4: Maximum Fruits into Baskets — Medium

Day 4.5: No Repeat Substring — Hard

Day 4.6: Sliding Window Maximum — Hard

End of Day 4

(Go to Top)

Maths

In technical interviews, the interviewer may ask problems based on basic maths like Fibonacci, some graph theory, probability theory and others that apply to Computer science.

Problems:

Day 5.1: Minimum Steps to Make Product equal to one — Easy

Day 5.2: Trailing Zeros in Factorial — Easy

Day 5.3: Find Fraction

Day 5.4: Count of Sum of Consecutive Natural Numbers

Day 5.5: Find the Number of Combinations given n and r. (nCr)

Day 5.6: Nth Natural Number

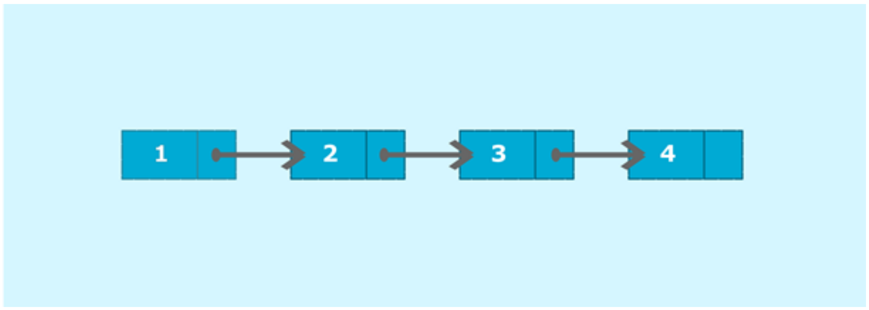
Day 5.7: Smallest Positive Integer that cannot be represented as Sum of Elements of the given array.

You are given an array with integers and a number N. You have to find the smallest integer that cannot be represented as the sum of elements of the array. #TODO .

End of Day 5

(Go to Top)

Linked List



If you have given enough interviews, you already know the importance of being good with problems on linked lists. My interview experience with Amazon and others experience as well tells me that they are obsessed with this topic .

Problems:

Day 6.1: Reverse a Linked List

Day 6.2: Delete a Node without the head pointer

Day 6.3: Add two numbers represented by linked lists

Day 6.4: Check if Linked List is a Palindrome

Day 6.5: Remove Nth node from the back of the linked list

Day 6.6: Find the intersection point of Y Linked List

Day 6.7: Detect a cycle in a linked list

End of Day 6

(Go to Top)

Day 7.1: Reverse Every K-elements sublist

Day 7.2: Find the starting point of the loop of a linked list

Day 7.3: Flattening of a linked list

Day 7.4: Rotate a Linked list

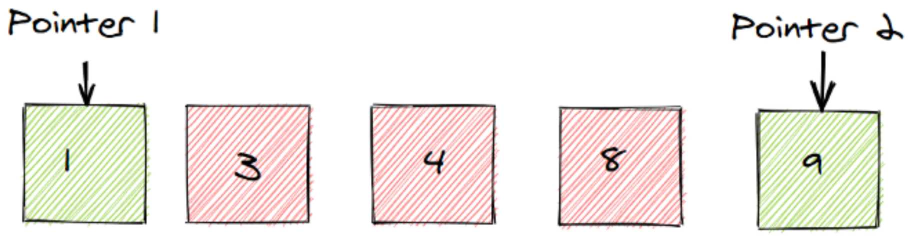
Day 7.5: Clone a Linked List with next and random pointers

Day 7.6: Reorder List

End of Day 7

(Go to Top)

2 Pointer



Problems:

Day 8.1: Two Sum — Easy

Day 8.2: Backspace String compare — Easy

Day 8.3: Find the Duplicate Number

Day 8.4: Three Some — Medium

Day 8.5: Three Some Closest — Medium

Day 8.6: Subarrays with product less than K — Medium

Day 8.7: Container with most Water — Medium

Optional

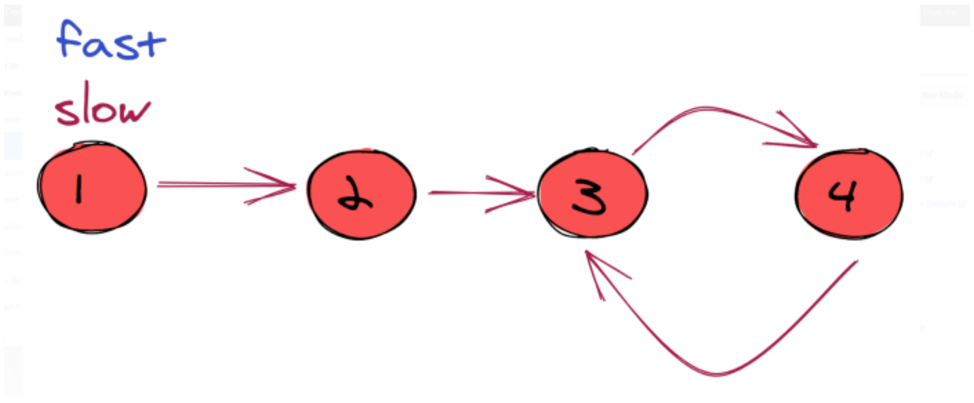
Max Consecutive Ones — Medium

Trapping Rain Water — Hard

End of Day 8

(Go to Top)

Fast and Slow pointers



Fast and slow pointers is a technique that is used to solve problems based on linked lists and arrays. The fast and slow pointers move at different speeds and it can be proven that they are bound to meet. Let's see how this approach works.

Problems:

Day 9.1: Middle of a Linked List — Easy

Day 9.2: Palindrome Linked List — Easy

Day 9.3: Remove Duplicates from Sorted List — Easy

Day 9.4: Linked List Cycle 2 — Medium

Day 9.5: Remove Nth Node from the End of List — Medium

Day 9.6: Sort List — Medium

Day 9.7: Cycle in a Circular Array — Hard

End of Day 9

(Go to Top)

Stack

Day 10.1: Implement Stack

Day 10.2: LRU Cache (Very Very Important)

Day 10.3: Parentheses Checker

Day 10.4: Next Larger Element

Day 10.5: The Celebrity Problem

Day 10.6: Remove K Digits

End of Day 10

(Go to Top)

Queue (2 Easy — 4 Medium — 1 Hard)

Problems:

Day 11.1: Implement Queue

Day 11.2: Largest Rectangle in Histogram

Day 11.3: First Negative Integer in Every Window of Size K

Day 11.4: Valid Substring

Day 11.5: Maximum of all subarrays of size k

Day 11.6: Circular Tour

Day 11.7: First Non-Repeating Character in a Stream

Optional

Rotten Orange (BFS)

End of Day 11

(Go to Top)

Hashing (3 Easy — 5 Medium)

Problems:

Day 12.1: Top K Frequent Elements in Array

Day 12.2: The intersection of two Arrays

Day 12.3: Triplet sum in Array

Day 12.4: Print Anagrams together

Day 12.5: Subarrays with Sum K

Day 12.6: Length of longest Substring

Day 12.7: Largest Subarray with 0 Sum

Day 12.8: Count number of subarrays with given XOR

Optional

Longest Substring with No Repeat

2 Sum Problem

4 Sum Problem

End of Day 12

(Go to Top)

Heap

Problems:

Day 13.1: Binary Heap Operations

Day 13.2: Minimum Cost of Ropes

Day 13.3: Adding Array Elements

Day 13.4: Kth Smallest element in a sorted Matrix

Day 13.5: Find K Pairs with Smallest Sums

Day 13.6: Meeting Rooms 2

Day 13.7: K closest point to Origin

Day 13.8: Top K Frequent Elements

End of Day 13

(Go to Top)

Day 14.1: Sort Characters by Frequency

Day 14.2: Merge K Sorted Lists

Day 14.3: Nearly Sorted

Day 14.4: Median of Stream

Day 14.5: Smallest Range covering Elements from K Lists

Day 14.6: Employee Free Time

Optional

Maximum Frequency Stack

Sliding Window Median

End of Day 14

(Go to Top)

Greedy

Problems:

Day 15.1: Maximize Toys

Day 15.2: Largest Number with Given Sum

Day 15.3: Jump Game

Day 15.4: Task Scheduler

Day 15.5: Reorganize String

Day 15.6: Coin Piles

Day 15.7: N Meetings in one Room

Day 15.8: Job Sequencing

Optional

Police and Thieves

Water the plants

Course Schedule — Hard

End of Day 15

(Go to Top)

Recursion (2 Easy — 7 Medium)

Problems:

Day 16.1: Number of Paths

Day 16.2: Juggler Sequence

Day 16.3: Permutations (All Parts)

Day 16.4: Pascals Triangle

Day 16.5: Subset Sums (All Parts)

Day 16.6: Subset-2 — Medium

Day 16.7: Combination Sum (All parts)

Day 16.8: Special Keyboard

Day 16.9: Rat in a maze

Optional

Generate Parenthesis

Flood Fill Algorithm

Word Boggle

End of Day 16

(Go to Top)

Backtracking (3 Medium — 3 Hard)

Problems:

Day 17.1: Palindrome Partitioning

Day 17.2: N Queen Problem

Day 17.3: Generalized Abbreviation

Day 17.4: M Coloring Problem

Day 17.5: Letter Combination of a Phone Number

Day 17.6: Sudoku Solver

End of Day 17

(Go to Top)

Divide & Conquer (5 Medium)

Problems:

Day 18.1: Matrix Median

Day 18.2: Find Element that appears once in a sorted array and rest appears twice

Day 18.3: Search Element in a sorted and rotated Array and Find the Pivot Where it is rotated

Day 18.4: Median of 2 Sorted Arrays

Day 18.5: K-th Element of Two Sorted Arrays

End of Day 18

(Go to Top)

String (2 Easy — 9 Medium)

Problems:

Day 19.1: Reverse words in a given String

Day 19.2: Roman Number to Integer

Day 19.3: Longest Common Prefix

Day 19.4: Length of Longest Prefix Suffix

Day 19.5: Implement Atoi

Day 19.6: Validate an IP address

Day 19.7: Look and say Pattern

Day 19.8: Longest substring without repeating characters

Optional

Longest K unique characters Substring

Longest Common Prefix

Rabin Karp

End of Day 19

(Go to Top)

Binary Tree

Problems:

Day 20.1: ZigZag Tree Traversal

Day 20.2: Checked for Balanced Tree

Day 20.3: Height of Binary Tree

Day 20.4: Diameter of Binary Tree

Day 20.5: Tree Traversals (Inorder — Preorder — Postorder)

Day 20.6: Connect nodes at the same level

Day 20.7: Sum tree

Day 20.8: Different Views of a Tree (Left View — Bottom View — Top View)

End of Day 20

(Go to Top)

Day 21.1: LCA in Binary Tree

Day 21.2: Check if two trees are identical or not

Day 21.3: Maximum Path Sum

Day 21.4: Construct Binary Tree from inorder and preorder

Day 21.5: Construct Binary Tree from inorder and postorder

Day 21.6: Flatten Binary Tree to Linked List

Day 21.7: Check if Binary Tree is a mirror of itself or not

Optional

Burning Tree

End of Day 21

(Go to Top)

Binary Search Tree

Problems:

Day 22.1: Check for BST

Day 22.2: Array to BST

Day 22.3: Inorder Successor in BST

Day 22.4: Populate Next right pointers of Tree

Day 22.5: Find LCA of two nodes in BST

Day 22.6: Find the inorder predecessor/successor of a given key in BST

Day 22.7: Floor and Cell in a BST

Day 22.8: Unique BSTs

Day 22.9: Serialize and Deserialize Binary Tree

Optional

Pair with a given target in BST

Merge two BSTs

Fixing two nodes of a BST

Sorted Linked List to BST

BST iterator

End of Day 22

(Go to Top)

Mixed Topics

Problems:

Day 23.1: Binary Tree to Doubly Linked List

Day 23.2: Find Median in a stream of Running Integers

Day 23.3: Employee Free Time

Day 23.4: Distinct Numbers in a Window

Day 23.5: Flood-Fill Algorithm

Day 23.6: Kth Largest Element in an unsorted array

End of Day 23

(Go to Top)

Graph

Problems:

Day 24.1: DFS

Day 24.2: BFS

Day 24.3:Count the Paths

Day 24.4:Eulerian Path in an Undirected Graph

Day 24.5:Number of Islands

Day 24.6:Covid Spread

Day 24.7: Clone a graph

Day 24.8: Number of Connected Components in an Undirected Graph

Day 24.9: Detect a Cycle in a Graph

Day 24.10: Topological sort

Optional (These are good to know problems)

Course Schedule (all versions are important)

Bipartite Check

Strongly Connected Components (SCC)

Dijkstra's Algorithm

Bellmann Ford Algorithm

Floyd Warshall Algorithm

MST Using Prim's Algorithm

MST Using Kruskal's Algorithm

End of Day 24

(Go to Top)

Dynamic Programming — (5 Easy & 22 Medium)

Dynamic Programming is not hard!; you just need to see and solve many problems. All you need is exposure. So learn about as many DP questions as possible.

I am sure; after learning these 27 problems, you will start getting DP .

Problems:

Day 25.1: Count Ways to Reach the nth Stair (Climbing stairs)

Day 25.1: Best Time to Buy and Sell Stock (This problem can have many versions ranging from easy to hard, read the article for all the versions.)

Day 25.1: Maximum Subarray

Day 25.2: Range Sum Query (Immutable)

Day 25.3: Counting bits

Day 25.4: Longest Increasing Subsequence

Day 25.5: Longest Common Subsequence

Day 25.6: Longest Palindromic Substring

Day 25.7: 0–1 Knapsack

Day 25.8: Edit Distance

Day 25.9: Maximum Sum increasing subsequence

Day 25.10: Matrix Chain Multiplication

End of Day 25

(Go to Top)

Day 26.1: House Robber

Day 26.2: Coin Change

Day 26.3: Maximum Product Subarray

Day 26.4: Word Break

Day 26.5: Decode Ways

Day 26.6: Unique Paths

Day 26.7: Palindromic Substrings

Day 26.8: Palindromic Partitioning

Day 26.9: Rod Cutting

Day 26.10: Egg Dropping

Optional

Player With Maximum Score

Matrix Chain Multiplication

Partition Equal Subset Sum

Partition to K equal sum subsets.

Maximum Profit in Job Scheduling

End of Day 26

(Go to Top)

Design Patterns — (3 Easy, 5 Medium & 1 Hard)

Problems:

Day 27.1: Stack Using Two Queues

Day 27.2: Queue Using Stack

Day 27.3: Queue Operations

Day 27.4: Implement a Trie (Prefix Tree)

Day 27.5: Serialize and Deserialize Binary Tree

Day 27.6: LRU Cache (Very Important)

Day 27.7:Binary Heap Operations

Day 27.8: Ternary Search

End of Day 27

(Go to Top)

Bits Manipulation — (4 Easy & 1 Medium)

Questions on bit manipulation are not that frequent in interviews but still, there's a chance. So if you want you can do this now or after every other topic, **If you still have time left for your interview.**

Problems:

Find MSB in Constant time.

Check if the Number is a Power of 2 in Constant Time

Counting Bits

Single Number

Power Set

End of Optional

(Go to Top)

Theory (OS — CN — DB)

- OS Notes
- CN Notes
- DB Notes

Behaviour Interview prep.

- How to prepare for your Behavioural interview and Why?
- #ToDo
- #ToDo

Project

- How should you present your Projects in an interview?

Under construction

I am working and posting articles every day to complete this list as soon as possible.

If you find this article helpful, clap for me .

Want to thank me?

(Go to Top)

Coding Interviews Best Interview Tips Behavioural Interviews Preparation Plan Faang