

Google DSA Previous Questions

Easy

- Write the code for this: **Logger Rate Limiter**
(Build a logger that only allows messages to be printed if they haven't been printed in the last 10 seconds.)
- Write the code for this: **Two Sum**
(Given an array and a target value, find two numbers that add up exactly to the target.)
- Write the code for this: **Moving Average from Data Stream**
(Implement a class that always returns the average of the last k values seen in a data stream.)
- Write the code for this: **Employee Importance**
(Given employee records (id, importance, subordinates), compute the total importance of one employee plus all of their reports.)
- Write the code for this: **Add to Array-Form of Integer**
(Add a given integer to a large number represented as an array of digits, and return the resulting array.)
- Write the code for this: **Jewels and Stones**
(Count how many characters in string S ("stones") are also in string J ("jewels").)

- Write the code for this: Palindrome Number
(Determine if an integer reads the same forwards and backwards.)
- Write the code for this: Max Consecutive Ones
(Return the length of the longest consecutive run of 1's in a binary array.)
- Write the code for this: First Bad Version
(Given an API `isBadVersion(v)`, find the first version in `[1...n]` marked bad using minimal calls.)
- Write the code for this: Merge Two Sorted Lists
(Merge two sorted linked lists into one sorted linked list.)
- Write the code for this: Range Sum Query – Immutable
(Preprocess an integer array so that you can quickly return the sum of any subarray `[i...j]`.)
- Write the code for this: Read N Characters Given Read4
(Implement a function `read(n)` using the provided `read4(buf)` API that reads 4 chars at a time.)
- Write the code for this: Backspace String Compare
(Given two strings where '#' means backspace, determine if they end up the same.)

- Write the code for this: Add Two Numbers
(Add two non-negative integers represented by linked lists, digit by digit, and return the sum list.)
 - Write the code for this: Majority Element
(Find the element that appears more than $\lfloor n/2 \rfloor$ times in an array.)
-

Medium

- Write the code for this: Longest Continuous Subarray With Absolute Diff $\leq$ Limit
(Find the length of the longest subarray where the difference between max and min is at most a given limit.)
- Write the code for this: Validate Stack Sequences
(Given two sequences pushed and popped, check if they could come from valid stack operations.)
- Write the code for this: Maximum Points You Can Obtain from Cards
(Pick exactly k cards from either end of an array to maximize your total score.)
- Write the code for this: Expressive Words
(Count how many words can be stretched (by repeating letters) to match a given target string.)

- Write the code for this: Find And Replace in String
(Given a string S, and arrays of indices, sources, targets—replace each source at the index if it matches.)
- Write the code for this: Missing Element in Sorted Array
(In a strictly increasing arithmetic array, find the k-th missing number.)
- Write the code for this: Find First and Last Position of Element in Sorted Array
(Return the start and end indices of a target value in a sorted array, or [-1,-1] if not found.)
- Write the code for this: Decode Ways
(Count how many ways a digit string can be decoded into letters (1→‘A’ ... 26→‘Z’).)
- Write the code for this: Split Array into Consecutive Subsequences
(Check if you can split an array into one or more subsequences of consecutive integers, each of length ≥ 3 .)
- Write the code for this: Alphabet Board Path
(Given a 5×6 letter board, output the sequence of moves (U/D/L/R/‘!’) to spell a target string.)

- Write the code for this: Subarray Sum Equals K
(Count the number of contiguous subarrays whose sum equals a given k.)
- Write the code for this: Number of Matching Subsequences
(Given a string and a list of words, count how many words are subsequences of the string.)
- Write the code for this: Decode String
(Decode an encoded string with numbers and brackets, e.g. "3[a]2[bc]" → "aaabcbc".)
- Write the code for this: Reorganize String
(Rearrange the characters so no two identical letters are adjacent; return any valid arrangement.)
- Write the code for this: Campus Bikes II
(Given workers and bikes on a 2D grid, assign each worker one bike to minimize total Manhattan distance.)
- Write the code for this: Online Election
(Track the current leading candidate over time given an array of votes with timestamps.)
- Write the code for this: Count of Smaller Numbers After Self
(For each element in an array, count how many smaller elements appear to its right.)

- Write the code for this: Find Duplicate Subtrees
(Return all root nodes of duplicate subtrees in a binary tree.)
- Write the code for this: Minimum Area Rectangle II
(Given a set of points, find the rectangle of any orientation with minimum area.)
- Write the code for this: Find Duplicate File in System
(Given file paths with file content, group all files with identical content.)
- Write the code for this: RLE Iterator
(Implement an iterator over a run-length encoded list: [count, value, count, value,...].)
- Write the code for this: Hand of Straights
(Check if you can rearrange an array of cards into groups of size k, each consecutive.)
- Write the code for this: Delete Nodes And Return Forest
(Given a binary tree and a list of nodes to delete, return the forest of remaining trees.)
- Write the code for this: Divide Array in Sets of K Consecutive Numbers

(Check if the array can be partitioned into sets of k consecutive numbers.)

- Write the code for this: Sequence Reconstruction
(Determine if the original sequence can be uniquely reconstructed from a list of subsequences.)
- Write the code for this: Random Pick with Weight
(Pick an index at random where the probability is proportional to a given weight array.)
- Write the code for this: Flip Equivalent Binary Trees
(Check if two binary trees are equivalent when you can swap any node's left and right children.)
- Write the code for this: My Calendar I
(Implement a calendar that supports booking events without any overlap.)
- Write the code for this: Campus Bikes
(Assign each worker one bike to minimize the total Manhattan distance (one-to-one matching).)
- Write the code for this: Split Array Largest Sum
(Split an array into m non-empty continuous subarrays so that the largest sum among them is minimized.)

- Write the code for this: Sliding Window Maximum
(Given an array and window size k, return an array of maximums in each sliding window.)
- Write the code for this: Top K Frequent Elements
(Return the k most frequent elements in an integer array.)
- Write the code for this: Candy Crush
(Simulate crushing any group of ≥ 3 adjacent identical candies and collapsing the board until stable.)
- Write the code for this: Invert Binary Tree
(Flip a binary tree left-to-right.)
- Write the code for this: Longest Substring with At Most Two Distinct Characters
(Find the length of the longest substring containing at most two different characters.)
- Write the code for this: Count Square Submatrices with All Ones
(Count all square submatrices of 1's in a binary matrix.)
- Write the code for this: Max Area of Island
(Return the area of the largest island (connected 1's) in a 2D grid.)

- Write the code for this: Valid Sudoku
(Check if a 9×9 Sudoku board is valid according to rules.)
- Write the code for this: Word Break
(Determine if a string can be segmented into space-separated dictionary words.)
- Write the code for this: Compare Version Numbers
(Compare two version strings like “1.0.1” vs “1”).)
- Write the code for this: 3Sum Smaller
(Count the number of triplets with sum less than a target.)
- Write the code for this: Number of Atoms
(Parse a chemical formula string and count the number of each atom.)
- Write the code for this: Word Ladder
(Find the minimum steps to transform one word to another by changing one letter at a time.)
- Write the code for this: Evaluate Reverse Polish Notation
(Evaluate an expression in Reverse Polish (postfix) notation.)

- Write the code for this: Count and Say
(Generate the nth term of the count-and-say sequence.)
- Write the code for this: Merge k Sorted Lists
(Merge k sorted linked lists into one sorted list.)
- Write the code for this: Remove Invalid Parentheses
(Return all valid strings by removing the minimum number of invalid parentheses.)
- Write the code for this: Design Hit Counter
(Count the number of hits received in the past 5 minutes.)
- Write the code for this: Course Schedule II
(Given prerequisites, return an order to take courses or detect impossibility.)
- Write the code for this: Shuffle an Array
(Implement an algorithm to randomly shuffle an array uniformly.)
- Write the code for this: Range Sum Query – 2D Immutable
(Preprocess a 2D matrix so you can quickly query the sum of any submatrix.)

- Write the code for this: Number of Closed Islands
(Count islands of 0's completely surrounded by 1's in a grid.)
- Write the code for this: Unique Paths
(Count the number of paths from top-left to bottom-right of a grid, moving only right or down.)
- Write the code for this: Combinations
(Generate all combinations of k numbers chosen from 1 to n.)
- Write the code for this: Populating Next Right Pointers in Each Node
(Connect each node to its immediate right neighbor in a perfect binary tree.)
- Write the code for this: Minimum Path Sum
(Given a grid of non-negative numbers, find a path from top-left to bottom-right with minimum sum.)
- Write the code for this: Generate Parentheses
(Generate all well-formed parentheses strings of length $2n$.)
- Write the code for this: Add Strings
(Add two non-negative integers represented as strings and return the sum as a string.)

- Write the code for this: Longest Consecutive Sequence
(Find the length of the longest sequence of consecutive integers in an array.)
- Write the code for this: Unique Binary Search Trees
(Count how many structurally unique BSTs can be built with n nodes.)
- Write the code for this: Pascal's Triangle
(Generate the first n rows of Pascal's triangle.)
- Write the code for this: Happy Number
(Determine if a number eventually reaches 1 when you repeatedly replace it by the sum of the squares of its digits.)
- Write the code for this: Plus One
(Given an array of digits representing an integer, add one to the integer and return the resulting digits array.)

Hard

- Write the code for this: Guess the Word
(Find a secret word by making ≤ 10 guesses; each guess returns the count of matching letters.)
- Write the code for this: Minimum Window Subsequence
(Find the shortest substring of S such that T is a subsequence of that substring.)

- Write the code for this: String Transforms Into Another String
(Determine if you can convert one string into another using wildcard character swaps without conflicts.)
- Write the code for this: Find Two Non-overlapping Subarrays Each With Target Sum
(Find two disjoint subarrays that each sum to a target, while minimizing their combined length.)
- Write the code for this: Cracking the Safe
(Generate the shortest string that contains every possible k-digit code (0–9) as a substring.)
- Write the code for this: Tiling a Rectangle with the Fewest Squares
(Cover an $m \times n$ rectangle exactly with the fewest integer-sided squares.)
- Write the code for this: Range Module
(Implement a data structure supporting adding, querying, and removing intervals.)
- Write the code for this: Word Abbreviation
(Generate the shortest unique abbreviation for each word in a list.)
- Write the code for this: Encode String with Shortest Length
(Compress a string by encoding repeated substrings to minimize total length.)

- Write the code for this: Divide Chocolate
(Split chocolate bars fairly under optimal play to maximize your share.)
- Write the code for this: Student Attendance Record II
(Count how many attendance records of length n are rewardable under given conditions.)
- Write the code for this: Minimum Number of K Consecutive Bit Flips
(Flip subarrays of length k to turn all bits to 1; return minimum flips or -1 .)
- Write the code for this: LFU Cache
(Design a cache that evicts the least-frequently used item when capacity is exceeded.)
- Write the code for this: Minimum Insertion Steps to Make a String Palindrome
(Compute the minimum insertions needed to turn a string into a palindrome.)
- Write the code for this: Regular Expression Matching
(Implement regex matching with support for '.' (any char) and '*' (zero or more).)*
- Write the code for this: Median of Two Sorted Arrays
(Find the median of two sorted arrays in $O(\log \min(m,n))$ time.)

- Write the code for this: Stone Game III
(Determine the optimal score difference in a turn-based stone game.)
- Write the code for this: Design Search Autocomplete System
(Given a set of historical sentences and frequencies, return top 3 matches for a typed prefix.)
- Write the code for this: Serialize and Deserialize N-ary Tree
(Convert an N-ary tree to a string and back.)
- Write the code for this: Minimum Number of Refueling Stops
(Compute the minimum number of gas stops needed to reach a target distance.)
- Write the code for this: Self Crossing
(Given moves in the plane, determine if the path crosses itself.)
- Write the code for this: Smallest Sufficient Team
(Select the smallest team of people whose combined skills cover all required skills.)
- Write the code for this: Max Value of Equation
(Given points (x_i, y_i) , maximize $y_i + y_j + |x_i - x_j|$ subject to $x_j - x_i \leq k$.)

- Write the code for this: Dungeon Game
(Compute the minimum initial health required to survive a dungeon grid with damage/heal cells.)
- Write the code for this: Edit Distance
(Find the minimum number of insert, delete, replace operations to convert word1 into word2.)
- Write the code for this: Frog Position After T Seconds
(On a tree, a frog starts at node 1 and jumps randomly; find probability it's on target after t seconds.)
- Write the code for this: Maximum Score Words Formed by Letters
(Given letter inventory and word list with scores, pick words to maximize total score.)
- Write the code for this: Minimum Number of Taps to Open to Water a Garden
(Given taps at positions with ranges, open the minimum taps to water an interval $[0, n]$.)
- Write the code for this: Strong Password Checker
(Determine minimum changes to make a password valid under length, character-type, and repeat rules.)

- Write the code for this: Alien Dictionary
(Given a sorted list of words in an alien language, deduce the alphabetical order of characters.)
- Write the code for this: Find Median from Data Stream
(Design a structure that supports inserting numbers and finding the current median.)
- Write the code for this: Web Crawler
(Crawl all URLs reachable from a start URL that share the same hostname.)
- Write the code for this: Sliding Window Median
(Given an array and window size k , return the median of each sliding window.)
- Write the code for this: Fruit Into Baskets
(Pick fruits into two baskets to maximize the number of fruits such that only two types are collected.)
- Write the code for this: Cherry Pickup II
(Two robots collect cherries from top to bottom of a grid; maximize total cherries collected.)

- Write the code for this: Cherry Pickup
(Traverse a grid twice (go and return) to pick up maximum cherries without counting twice.)
- Write the code for this: Maximum Profit in Job Scheduling
(Given jobs with start, end, and profit, schedule non-overlapping jobs to maximize profit.)
- Write the code for this: Course Schedule III
(Maximize the number of courses you can take before their deadlines.)
- Write the code for this: Binary Tree Cameras
(Place the fewest cameras in a tree so that every node is monitored.)
- Write the code for this: Text Justification
(Given a list of words and a max width, fully justify text line by line.)
- Write the code for this: Letter Combinations of a Phone Number
(Given digit string [2–9], return all possible letter combinations on a phone keypad.)
- Write the code for this: Longest Increasing Path in a Matrix
(Find the longest path of strictly increasing values in a 2D grid.)
- Write the code for this: Integer to English Words
(Convert a non-negative integer to its English words representation.)

- Write the code for this: Minimum Window Substring
(Find the smallest substring of S that contains all characters of T.)
- Write the code for this: All O'one Data Structure
(Design a data structure supporting inc/dec/getMaxKey/getMinKey in $O(1)$ time.)
- Write the code for this: Merge k Sorted Lists
(Merge k sorted linked lists into one sorted linked list.)
- Write the code for this: LFU Cache
(Design a cache that evicts the least-frequently used key when capacity is full.)
- Write the code for this: Course Schedule II
(Given course prerequisites, return any valid ordering to finish all courses.)
- Write the code for this: Range Sum Query – 2D Immutable
(Preprocess a matrix to support querying sums of any submatrix efficiently.)
- Write the code for this: Cracking the Safe
(Generate a shortest string that contains every possible k-digit password exactly once.)

- Write the code for this: Serialize and Deserialize N-ary Tree
(Write functions to encode an N-ary tree to a string and decode it back.)
- Write the code for this: Tiling a Rectangle with the Fewest Squares
(Cover a given rectangle exactly using the fewest integer-sided square tiles.)
- Write the code for this: Regular Expression Matching
(Implement a regex matcher with support for '.' and '*' operators.)*
- Write the code for this: Median of Two Sorted Arrays
(Find the median of two sorted arrays in $O(\log \min(m,n))$ time.)
- Write the code for this: LFU Cache
(Implement a cache with get/put operations and evict the least-frequently used item.)
- Write the code for this: Merge k Sorted Lists
(Merge k sorted linked lists into one.)
- Write the code for this: Minimum Window Subsequence
(Find the shortest substring of S that still contains T as a subsequence.)

- Write the code for this: Cracking the Safe
(Construct the shortest de Bruijn sequence containing every k-length combination.)
-

CodeNeuraX