🧙 Leetcode Wizard

🟢 Easy Problems

Problem Name	Short Problem Description
Two Sum (#1)	Find indices of two numbers in an array that add up to a specific target.
Roman to Integer (#13)	Convert a Roman numeral string to its integer value.
Valid Parentheses (#20)	Determine if the input string has valid matching parentheses.
Same Tree (#100)	Check if two binary trees are identical in structure and node values.
Symmetric Tree (#101)	Determine if a binary tree is symmetric around its center.
Best Time to Buy and Sell Stock (#121)	Find the maximum profit from buying and selling a stock once.
Diameter of Binary Tree (#543)	Find the diameter (longest path) of a binary tree.

🟠 Medium Problems

Problem Name	Short Problem Description
Longest Palindromic Substring (#5)	Find the longest substring in a given string that is a palindrome.
String to Integer (atoi) (#8)	Convert a string to a 32-bit signed integer following specific parsing rules.
Integer to Roman (#12)	Convert an integer to its corresponding Roman numeral representation.
Valid Sudoku (#36)	Check if a 9x9 Sudoku board is valid according to Sudoku rules.
Combination Sum (#39)	Find all unique combinations in an array that sum up to a target value.
Permutations (#46)	Generate all possible permutations of a list of numbers.

Rotate List (#61)	Rotate a linked list to the right by k places.
Minimum Path Sum (#64)	Find the minimum sum path from top-left to bottom-right in a grid.
Word Search (#79)	Determine if a given word exists in a grid of letters by traversing adjacent cells.
Validate Binary Search Tree (#98)	Verify if a binary tree is a valid binary search tree (BST).
Binary Tree Level Order Traversal (#102)	Return the level order traversal of a binary tree's nodes' values.
Convert Sorted List to Binary Search Tree (#109)	Convert a sorted singly linked list to a height-balanced BST.
Populating Next Right Pointers in Each Node (#116)	Populate each next pointer to point to its next right node in a perfect binary tree.
LRU Cache (#146)	Design and implement a Least Recently Used (LRU) cache.
Min Stack (#155)	Design a stack that supports push, pop, top, and retrieving the minimum element in constant time.
Number of Islands (#200)	Count the number of islands in a 2D grid representing land and water.
Course Schedule (#207)	Determine if all courses can be finished given their prerequisites.
Course Schedule II (#210)	Return the order in which courses should be taken to finish all courses.
Add and Search Word - Data structure design (#211)	Design a data structure to add words and search for them with support for wildcard characters.
Game of Life (#289)	Simulate the Game of Life rules on a given board.
Longest Increasing Subsequence (#300)	Find the length of the longest increasing subsequence in an array.

Pacific Atlantic Water Flow (#417)	Find all coordinates where water can flow to both the Pacific and Atlantic oceans.
Minesweeper (#529)	Implement the Minesweeper game logic to reveal cells based on clicks.
Reorganize String (#767)	Rearrange the characters in a string so that no two adjacent characters are the same.

🔴 Hard Problems

Problem Name	Short Problem Description
Median of Two Sorted Arrays (#4)	Find the median of two sorted arrays combined into one sorted array.
Merge K Sorted Lists (#23)	Merge k sorted linked lists into one sorted linked list.
Word Ladder (#127)	Find the length of the shortest transformation sequence from a start word to an end word.
Word Ladder II (#126)	Find all shortest transformation sequences from a start word to an end word.
Word Search II (#212)	Find all words from a dictionary that exist in a 2D grid of letters.
Integer to English Words (#273)	Convert a non-negative integer to its English words representation.
Find Median from Data Stream (#295)	Continuously find the median of a stream of numbers.