Section 1: Subsets and Combinations (Easy)

These problems build your foundation for exploring all possible selections from a set.

Q1. Subsets Given an array of distinct integers, return all possible subsets (the power set). The solution set must not contain duplicate subsets. 🔗 LeetCode: https://leetcode.com/problems/subsets/

Q2. Subsets II (with duplicates) Same as Q1 but the input array may contain duplicates. How do you avoid duplicate subsets in the result? 🔗 LeetCode: https://leetcode.com/problems/subsets-ii/

Q3. Combination Sum III Find all combinations of k numbers that sum to n, using only numbers 1 to 9, each used at most once. 🔗 LeetCode: https://leetcode.com/problems/combination-sum-iii/

Q4. Letter Combinations of a Phone Number Given a string of digits, return all possible letter combinations that the number could represent (like a phone keypad). 🔗 LeetCode: https://leetcode.com/problems/letter-combinations-of-a-phone-number/

Q5. Combination Sum (Unlimited Use) Find all combinations where chosen numbers sum to target. Each number can be used unlimited times. 🔗 LeetCode: https://leetcode.com/problems/combination-sum/

Section 2: Permutations and Arrangements (Medium)

These problems require generating all possible orderings or arrangements, making the "undo" step critical.

Q1. Permutations Given a collection of distinct integers, return all possible permutations. 🔗 LeetCode: https://leetcode.com/problems/permutations/

Q2. Permutations II (with duplicates) Same as Q1 but the array may contain duplicate numbers. Return only unique permutations. 🔗 LeetCode: https://leetcode.com/problems/permutations-ii/

Q3. Palindrome Partitioning Partition a string such that every substring of the partition is a palindrome. Return all possible partitions. 🔗 LeetCode: https://leetcode.com/problems/palindrome-partitioning/

Q4. Generate Parentheses Generate all combinations of well-formed parentheses given n pairs. 🔗 LeetCode: https://leetcode.com/problems/generate-parentheses/

Q5. Combination Sum II (each number used once) Find all unique combinations in an array where candidate numbers sum to a target. Each number may only be used once. 🔗 LeetCode: https://leetcode.com/problems/combination-sum-ii/

Section 3: Grid and Path Problems (Medium)

These problems apply backtracking on a 2D grid, where you explore paths and undo moves.

Q1. Rat in a Maze A rat starts at (0,0) in a maze and needs to reach (N-1, N-1). Find all possible paths. 🔗 GFG: https://www.geeksforgeeks.org/rat-in-a-maze-backtracking-2/

Q2. Word Search Given an m x n grid of characters, find if the word exists in the grid by searching horizontally and vertically. 🔗 LeetCode: https://leetcode.com/problems/word-search/

Q3. Word Search II Same as above but find all words from a given list that exist in the board. Use Trie + Backtracking. 🔗 LeetCode: https://leetcode.com/problems/word-search-ii/

Q4. Unique Paths III Walk over every non-obstacle square exactly once on a grid. Return the number of such paths. 🔗 LeetCode: https://leetcode.com/problems/unique-paths-iii/

Q5. Knight's Tour Problem Move a chess knight on an N x N board such that it visits every cell exactly once. 🔗 GFG: https://www.geeksforgeeks.org/the-knights-tour-problem-backtracking-1/

Section 4: Constraint Satisfaction Problems (Hard)

These are classic backtracking problems with strict rules that heavily rely on pruning.

Q1. N-Queens Place N queens on an N x N chessboard such that no two queens attack each other. Return all distinct solutions. 🔗 LeetCode: https://leetcode.com/problems/n-queens/

Q2. Sudoku Solver Write a program to solve a Sudoku puzzle by filling in the empty cells. The solution must be valid. 🔗 LeetCode: https://leetcode.com/problems/sudoku-solver/

Q3. M Coloring Problem Given an undirected graph, determine if it can be colored with at most M colors such that no two adjacent vertices share the same color. 🔗 GFG: https://www.geeksforgeeks.org/m-coloring-problem-backtracking-5/

Q4. Expression Add Operators Given a string of digits and a target, add operators +, -, or * between the digits to make the expression evaluate to the target. 🔗 LeetCode: https://leetcode.com/problems/expression-add-operators/

Q5. Remove Invalid Parentheses Remove the minimum number of invalid parentheses to make the input string valid. Return all possible results. 🔗 LeetCode: https://leetcode.com/problems/remove-invalid-parentheses/