### **Hashmap**

1. Given an array of integers, find the first repeating element.
2. Determine if two strings are anagrams of each other using a hashmap.
3. Count the frequency of each character in a string and return the results.
4. Find the longest substring without repeating characters.
5. Given an array of words, group the words that are anagrams.

### **Sorting and Searching**

1. Find the intersection of two arrays.
2. Determine if a given value exists in a sorted 2D matrix.
3. Given an unsorted array, find the median value.
4. Find all pairs in an array that sum to a specific target.
5. Given a list of scores, find the second highest score.

### **Tree Concepts**

1. Given a binary tree, determine its maximum depth.
2. Check if a binary tree is a valid binary search tree (BST).
3. Find all leaf nodes in a binary tree.
4. Determine the level order traversal of a binary tree.
5. Count the number of nodes in a complete binary tree.

### **Recursion**

1. Generate all permutations of a given string.
2. Solve the N-Queens problem using recursion.
3. Count how many ways you can climb to the top of a staircase with variable steps.
4. Write a function to determine if a string is a palindrome using recursion.
5. Find all combinations of a set of numbers that sum to a target value.

### **Divide and Conquer**

1. Find the maximum element in a rotated sorted array.
2. Determine the closest pair of points from a set of points using a divide and conquer approach.
3. Count the number of occurrences of a number in a sorted array.
4. Find the largest rectangle in a histogram.
5. Given two sorted arrays, find the median of the two sorted arrays.

### **Heap Data Structure**

1. Find the kth largest element in an unsorted array.
2. Given a list of tasks with their durations, determine the order to complete them using a heap.
3. Merge k sorted linked lists into one sorted linked list.
4. Implement a function to find the top k frequent elements in an array.
5. Create a function to determine the minimum cost to connect all ropes given their lengths.