

Two Pointers

Easy

1. [Two Sum](#)
2. [Merge Sorted Array](#)
3. [Backspace String Compare](#)
4. [Palindrome Linked List](#)
5. [Merge Two Sorted List](#)

Medium

1. [Longest Substring Without Repeating Characters](#)
2. [Longest Palindrome Substring](#)
3. [Container With Most Water](#)
4. [Remove Duplicates from Sorted List II](#)
5. [3Sum](#)
6. [Interval List Intersections](#)
7. [Find All Anagrams in a String](#)
8. [Subarray Product Less Than K](#)
9. [Minimum Size Subarray Sum](#)
10. [Remove Nth Node From End of List](#)
11. [3Sum Closest](#)
12. [Add Two Numbers II](#)

Hard

1. [Trapping Rain Water](#)
2. [Rotate Array](#)
3. [Minimum Window Substring](#)

Queues/Monotonic Queues/Priority Queues

Easy

1. [Rank Transform of an Array](#)

Medium

1. [K Closest Points to Origin](#)
2. [Binary Tree Level Order Traversal II](#)

Hard

1. [Sliding Window Maximum](#)
2. [Constrained Subsequence Sum](#)
3. [Minimum Cost to Hire K Workers](#)
4. [Merge k Sorted Lists](#)

Stacks/Monotonic Stacks

Easy

1. [Reverse Linked List](#)
2. [Valid Parentheses](#)

Medium

1. [Smallest Subsequence of Unique Characters](#)
2. [Min Stack](#)
3. [Basic Calculator II](#)
4. [Find the Most Competitive Subsequence](#)
5. [Reorder List](#)
6. [Daily Temperatures](#)
7. [Next Greater Element II](#)
8. [Asteroid Collision](#)
9. [Minimum Remove to Make Valid Parentheses](#)

Hard

1. [Longest Valid Parentheses](#)
2. [Create Maximum Number](#)
3. [Number of Atoms](#)
4. [Largest Rectangle in Histogram](#)

BFS/DFS/Shortest Paths

Easy

1. [Subtree of Another Tree \(KMP\)](#)
2. [Same Tree](#)
3. [Symmetric Tree](#)

Medium

1. [Cheapest Flights Within K Stops](#)
2. [Path With Minimum Effort](#)
3. [Course Schedule II](#)
4. [Implement Trie \(Trie\)](#)
5. [Number of Islands](#)
6. [Number of Provinces](#)
7. [Populating Next Right Pointers in Each Node](#)
8. [Shortest Path in Binary Matrix](#)
9. [All Paths From Source To Target](#)
10. [Rotten Oranges](#)
11. [N-ary Tree Level Order Traversal](#)
12. [Time Needed to Inform All Employees](#)
13. [Binary Tree Right Side View](#)
14. [Most Stones Removed with Same Row or Column](#)

Hard

1. [Bus Routes](#)

Recursion/Backtracking

Easy

1. [Balanced Binary Tree](#)
2. [Invert Binary Tree](#)
3. [Diameter of Binary Tree](#)
4. [Binary Tree Inorder Traversal](#)

Medium

1. [Permutations II](#)
2. [Generate Parentheses](#)
3. [Path Sum III](#)
4. [Kth Smallest Element in a BST](#)
5. [Binary Search Tree Iterator](#)
6. [Subsets](#)
7. [Subsets II](#)
8. [Permutations](#)
9. [Combination Sum](#)
10. [Combination Sum II](#)
11. [Letter Combinations of a Phone Number](#)
12. [Word Search](#)
13. [Linked List in Binary Tree](#)
14. [Validate Binary Search Tree](#)

Hard

Dynamic Programming

Easy

1. [Repeated Substring Pattern](#)
2. [Climbing Stairs](#)

Medium

1. [Jump Game II](#)
2. [House Robber II](#)
3. [Longest Increasing Subsequence](#)
4. [Number of Longest Increasing Subsequence](#)
5. [Partition Equal Subset Sum](#)
6. [Maximum Product Subarray](#)
7. [Generate Parentheses](#)
8. [Maximum Subarray](#)
9. [Jump Game](#)
10. [House Robber](#)
11. [Unique Paths](#)
12. [Arithmetic Slices](#)
13. [Word Break](#)
14. [Longest Common Subsequence](#)
15. [Delete Operation for Two Strings](#)
16. [Coin Change](#)
17. [Integer Break](#)

Hard

1. [Super Egg Drop](#)
2. [Edit Distance](#)

Greedy Algorithm

Easy

1. [Lemonade Change](#)

Medium

1. [Find Valid Matrix Given Row and Column Sums](#)
2. [Task Scheduler](#)
3. [Smallest Range II](#)
4. [Valid Parenthesis String](#)
5. [Non-overlapping Intervals](#)
6. [Increasing Triplet Subsequence](#)
7. [Partition Labels](#)

Hard

1. [Couples Holding Hands](#)
2. [Candy](#)

Miscellaneous

String

1. [Group Anagrams](#)
2. [Longest Substring with At Least K Repeating Characters](#)
3. [Longest Palindrome](#)

Linked List

1. [Intersection of Two Linked List](#)
2. [Odd Even Linked List](#)
3. [Copy List with Random Pointer](#)
4. [Linked List Cycle II](#)
5. [Swap Nodes in Pairs](#)
6. [Reverse Nodes in k-Group](#)

Design

1. [LRU Cache](#)
2. [Dinner Plate Stacks](#)
3. [Implement Queue Using Stacks](#)
4. [Implement Stack Using Queues](#)
5. [Design Circular Queue](#)
6. [Seat Reservation Manager](#)
7. [Design HashMap](#)

Math/Logic/Bit Manipulation

1. [Next Greater Element III](#)
2. [Single Number](#)
3. [Subarray Sum Equals K](#)
4. [Max Points on a Line](#)
5. [Integer Break](#)
6. [Bitwise AND of Numbers Range](#)

Others

1. [Shuffle an Array](#) (Fisher Yates Algorithm)
2. [Majority Element](#) (Boyer-Moore Algorithm)
3. [First Missing Positive](#) (Cycle Sort)
4. [Sort Colors](#) (Dutch National Flag Problem)
5. [Sort List](#) (Merge Sort)
6. [Merge Intervals](#)
7. [Monotonic Array](#)
8. [Range Sum Query 2D - Immutable](#)
9. [Insert Interval](#)

10. [Insert Delete GetRandom O\(1\)](#)
11. [Set Matrix Zeroes](#)
12. [Product of Array Except Self](#)
13. [Word Pattern](#)

Binary Search

1. [Search in Rotated Sorted Array](#)
2. [Median of Two Sorted Array](#)
3. [Search a 2D Matrix](#)
4. [Search a 2D Matrix II](#)
5. [Find Minimum in Rotated Sorted Array](#)
6. [Find Peak Element](#)
7. [My Calendar I](#)
8. [Find First and Last Position of Element in Sorted Array](#)