



# LeetCode 48 Must- Practice

Algorithm  
Questions



Explore is a well-organized tool that helps you get the most out of LeetCode by providing structure to guide your progress towards the next

## 1 Two Sum

**Problem:** Given an array of integers nums and an integer target return indices of the two numbers such that they add up to the target



[Practice Link](#)

## 2 Add Two Numbers

**Problem:** Add two non-empty linked lists representing two non-negative integers



[Practice Link](#)

## 3 Longest Unique Substring

**Problem:** Add two non-empty linked lists representing two non-negative integers



[Practice Link](#)

## 4 Median of Two Sorted Arrays

**Problem:** Find the median of two sorted arrays



[Practice Link](#)

## 5 Longest Palindromic Substring

**Problem:** Find the longest palindromic substring in a given string



[Practice Link](#)

## 6 Container With Most Water

**Problem:** Given  $n$  non-negative integers representing the heights, find two lines that form a container to hold the most water



[Practice Link](#)

## 7 Merge Two Sorted Lists

**Problem:** Merge two sorted linked lists into one sorted list



[PracticeLink](#)

## 8 Best Time to Buy and Sell Stock

**Problem:** Find the maximum profit by choosing a single day to buy one stock and a different day to sell



[PracticeLink](#)

## 9 Valid Parentheses

**Problem:** Determine if the input string of parentheses is valid



[Practice Link](#)

## 10 Search in Rotated Sorted Array

**Problem:** Search for a target value in a rotated sorted array



[PracticeLink](#)

## 11 Combination Sum

**Problem:** Find all unique combinations of candidates where the chosen numbers sum to the target



[PracticeLink](#)

## 12 Permutations

**Problem:** Return all possible permutations of a collection of numbers



[Practice Link](#)

## 13 Subsets

**Problem:** Return all possible subsets of a given array



[PracticeLink](#)

## 14 Word Search

**Problem:** Find if a word exists in a grid following specific rules



[PracticeLink](#)

## 15 Climbing Stairs

**Problem:** Calculate the number of distinct ways to reach the top of a staircase with `n` steps.



[PracticeLink](#)

## 16 Coin Change

**Problem:** Find the minimum number of coins needed to make a certain amount



[PracticeLink](#)

## 17 Longest Increasing Subsequence

**Problem:** Find the length of the longest strictly increasing subsequence



[PracticeLink](#)

## 18 Binary Tree Inorder Traversal

**Problem:** Perform in order traversal on a binary tree



[PracticeLink](#)

## 19 Maximum Depth of Binary Tree

**Problem:** Find the maximum depth of a binary tree



[PracticeLink](#)

## 20 Validate Binary Search Tree

**Problem:** Determine if a binary tree is a valid binary search tree



[PracticeLink](#)

## 21 Kth Smallest Element in a BST

**Problem:** Find the kth smallest element in a binary search tree



[PracticeLink](#)



## 22 Binary Tree Level Order Traversal

**Problem:** Return level-order traversal of a binary tree



[PracticeLink](#)

## 23 Serialize and Deserialize Binary Tree

**Problem:** Convert a binary tree into a serialized string and vice versa



[PracticeLink](#)

## 24 Lowest Common Ancestor of Binary Tree

**Problem:** Find the lowest common ancestor of two nodes in a binary tree



[PracticeLink](#)

## 25 Product of Array Except Self

**Problem:** Return an array `answer` where `answer[i]` is the product of all elements of the array except `nums[i]`



[PracticeLink](#)

## 26 Maximum Subarray

**Problem:** Find the contiguous subarray with the maximum sum



[PracticeLink](#)

## 27 Merge Intervals

**Problem:** Merge overlapping intervals



[PracticeLink](#)

## 28 Insert Interval

**Problem:** Insert a new interval into a list of non-overlapping intervals



[PracticeLink](#)

## 29 Non-overlapping Intervals

**Problem:** Find the minimum number of intervals to remove to make the rest of the intervals non-overlapping



[PracticeLink](#)

## 30 Search a 2D Matrix

**Problem:** Search for a target value in an  $m \times n$  matrix



[PracticeLink](#)

## 31 Valid Anagram

**Problem:** Determine if two strings are anagrams of each other



[PracticeLink](#)

## 32 Group Anagrams

**Problem:** Group strings that are anagrams of each other



[PracticeLink](#)

## 33 Minimum Window Substring

**Problem:** Find the minimum window substring in `s` that contains all the characters of `t`



[PracticeLink](#)

## 34 Sliding Window Maximum

**Problem:** Find the maximum in each sliding window of size `k`



[PracticeLink](#)

## 35 Top K Frequent Elements

**Problem:** Find the k most frequent elements in an array



[PracticeLink](#)

## 36 Number of Islands

**Problem:** Count the number of islands in a grid



[PracticeLink](#)

## 37 Word Search II

**Problem:** Find all words on the board



[PracticeLink](#)

## 38 Course Schedule

**Problem:** Determine if you can finish all courses given the prerequisites



[PracticeLink](#)

## 39 Implement Trie (Prefix Tree)

**Problem:** Implement a trie with `insert`, `search`, and `startsWith` functions



[PracticeLink](#)

## 40 Palindrome Partitioning

**Problem:** Partition a string such that every substring is a palindrome



[PracticeLink](#)