



# The Blind 75

# LEETCODE

---

## Challenge



Master these & Crack MAANG!

## **\*Disclaimer\***

Everyone learns uniquely.

What matters is developing the problem solving ability to solve new problems.

This Doc will help you with the same.

# ARRAY



## 1. Two Sum

Given an array of integers `nums` and an integer `target`, return indices of the two numbers such that they add up to `target`.

You may assume that each input would have exactly one solution, and you may not use the same element twice.

You can return the answer in any order.

[Practice](#)

## 2. Best Time to Buy and Sell Stock

You are given an array `prices` where `prices[i]` is the price of a given stock on the `i`th day.

You want to maximize your profit by choosing a single day to buy one stock and choosing a different day in the future to sell that stock.

Return the maximum profit you can achieve from this transaction. If you cannot achieve any profit, return 0.

[Practice](#)



## 3. Contains Duplicate

Given an integer array `nums`, return true if any value appears at least twice in the array, and return false if every element is distinct.

[Practice](#)

## 4. Product of Array Except Self

Given an integer array `nums`, return an array `answer` such that `answer[i]` is equal to the product of all the elements of `nums` except `nums[i]`.

The product of any prefix or suffix of `nums` is guaranteed to fit in a 32-bit integer.

You must write an algorithm that runs in  $O(n)$  time and without using the division operation.

[Practice](#)

## 5. Maximum Subarray

Given an integer array `nums`, find the subarray with the largest sum, and return its sum.

[Practice](#)



## 6. Maximum Product Subarray

Practice

## 7. Find Minimum in Rotated Sorted Array

Practice

## 8. Search in Rotated Sorted Array

Practice

## 9. 3 Sum

Practice

## 10. Container With Most Water

Practice



# BINARY

## 11. Sum of Two Integers

Given two integers  $a$  and  $b$ , return the sum of the two integers without using the operators  $+$  and  $-$ .

Practice

## 12. Number of 1 Bits

Write a function that takes the binary representation of an unsigned integer and returns the number of '1' bits it has (also known as the Hamming weight).

Note:

- Note that in some languages, such as Java, there is no unsigned integer type. In this case, the input will be given as a signed integer type. It should not affect your implementation, as the integer's internal binary representation is the same, whether it is signed or unsigned.

In Java, the compiler represents the signed integers using 2's complement notation. Therefore, in Example 3, the input represents the signed integer. -3.

Practice



## 13. Counting Bits

Given an integer  $n$ , return an array  $\text{ans}$  of length  $n + 1$  such that for each  $i$  ( $0 \leq i \leq n$ ),  $\text{ans}[i]$  is the number of 1's in the binary representation of  $i$ .

[Practice](#)

## 14. Missing Number

[Practice](#)

## 15. Reverse Bits

[Practice](#)



# DYNAMIC PROGRAMMING

## 16. Climbing Stairs

You are climbing a staircase. It takes  $n$  steps to reach the top.

Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

[Practice](#)

## 17. Coin Change

You are given an integer array `coins` representing coins of different denominations and an integer `amount` representing a total amount of money.

Return the fewest number of coins that you need to make up that amount. If that amount of money cannot be made up by any combination of the coins, return -1.

You may assume that you have an infinite number of each kind of coin.

[Practice](#)



# 18. Longest Increasing Subsequence

Given an integer array `nums`, return the length of the longest strictly increasing subsequence.

[Practice](#)

# 19. Longest Common Subsequence

Given two strings `text1` and `text2`, return the length of their longest common subsequence. If there is no common subsequence, return 0.

A subsequence of a string is a new string generated from the original string with some characters (can be none) deleted without changing the relative order of the remaining characters.

- For example, "ace" is a subsequence of "abcde".

A common subsequence of two strings is a subsequence that is common to both strings.

[Practice](#)



# 20. Word Break Problem

Given a string s and a dictionary of strings wordDict, return true if s can be segmented into a space-separated sequence of one or more dictionary words.

Note that the same word in the dictionary may be reused multiple times in the segmentation.

[Practice](#)

# 21. Combination Sum

Given an array of distinct integers nums and a target integer target, return the number of possible combinations that add up to target.

The test cases are generated so that the answer can fit in a 32-bit integer.

[Practice](#)



## 22. House Robber

Practice

## 23. House Robber II

Practice

## 24. Decode Ways

Practice

## 25. Unique Paths

Practice

## 26. Jump Game

Practice



# GRAPH

## 27. Longest Consecutive Sequence

Given an unsorted array of integers `nums`, return the length of the longest consecutive elements sequence.

You must write an algorithm that runs in  $O(n)$  time.

[Practice](#)

## 28. Course Schedule

There are a total of `numCourses` courses you have to take, labeled from 0 to `numCourses` - 1. You are given an array `prerequisites` where `prerequisites[i] = [ai, bi]` indicates that you must take course `bi` first if you want to take course `ai`.

- For example, the pair `[0, 1]`, indicates that to take course 0 you have to first take course 1.

Return true if you can finish all courses. Otherwise, return false.

[Practice](#)



## 29. Pacific Atlantic Water Flow

Practice



## 30. Number of Islands

Practice

## 31. Clone Graph

Practice

## 32. Alien Dictionary

Practice

## 33. Graph Valid Tree

Practice



# 34. Number of Connected Components in an Undirected Graph (Leetcode Premium)

Practice



# INTERVAL

## 35. Insert Interval

You are given an array of non-overlapping intervals `intervals` where `intervals[i] = [starti, endi]` represent the start and the end of the *i*th interval and `intervals` is sorted in ascending order by `starti`. You are also given an interval `newInterval = [start, end]` that represents the start and end of another interval.

Insert `newInterval` into `intervals` such that `intervals` is still sorted in ascending order by `starti` and `intervals` still does not have any overlapping intervals (merge overlapping intervals if necessary).

Return intervals after the insertion.

[Practice](#)

## 36. Merge Intervals

Given an array of intervals where `intervals[i] = [starti, endi]`, merge all overlapping intervals, and return an array of the non-overlapping intervals that cover all the intervals in the input.

[Practice](#)



# 37. Non-overlapping Intervals

Practice

# 38. Meeting Rooms

Practice

# 39. Meeting Rooms II

Practice



# LINKED LIST

## 40. Reverse a Linked List

Given the head of a singly linked list, reverse the list, and return the reversed list.

Practice

## 41. Detect Cycle in a Linked List

Given head, the head of a linked list, determine if the linked list has a cycle in it.

There is a cycle in a linked list if there is some node in the list that can be reached again by continuously following the next pointer. Internally, pos is used to denote the index of the node that tail's next pointer is connected to. Note that pos is not passed as a parameter.

Return true if there is a cycle in the linked list. Otherwise, return false.

Practice



## 42. Merge Two Sorted Lists

You are given the heads of two sorted linked lists `list1` and `list2`.

Merge the two lists into one sorted list. The list should be made by splicing together the nodes of the first two lists.

Return the head of the merged linked list.

[Practice](#)

## 43. Merge K Sorted Lists

[Practice](#)

## 44. Remove Nth Node From End Of List

[Practice](#)

## 45. Reorder List

[Practice](#)



# MATRIX

## 46. Set Matrix Zeroes

Given an  $m \times n$  integer matrix  $\text{matrix}$ , if an element is 0, set its entire row and column to 0's.

You must do it in place.

[Practice](#)

## 47. Spiral Matrix

Given an  $m \times n$  matrix, return all elements of the matrix in spiral order.

[Practice](#)

## 48. Rotate Image

[Practice](#)

## 49. Word Search

[Practice](#)



# STRING

## 50. Longest Substring Without Repeating Characters

Given a string s, find the length of the longest substring without repeating characters.

Practice

## 51. Longest Repeating Character Replacement

You are given a string s and an integer k. You can choose any character of the string and change it to any other uppercase English character. You can perform this operation at most k times.

Return the length of the longest substring containing the same letter you can get after performing the above operations.

Practice



## 52. Valid Parentheses

Practice

## 53. Valid Anagram

Practice

## 54. Group Anagrams

Practice

## 55. Minimum Window Substring

Practice

## 56. Valid Palindrome

Practice



## 57. Longest Palindromic Substring

Practice



## 58. Palindromic Substrings

Practice

## 59. Encode and Decode Strings

Practice



## 60. Maximum Depth of Binary Tree

Given the root of a binary tree, return its maximum depth.

A binary tree's maximum depth is the number of nodes along the longest path from the root node down to the farthest leaf node.

[Practice](#)

## 61. Same Tree

Given the roots of two binary trees p and q, write a function to check if they are the same or not.

Two binary trees are considered the same if they are structurally identical, and the nodes have the same value.

[Practice](#)

## 62. Invert/Flip Binary Tree

Given the root of a binary tree, invert the tree, and return its root.

[Practice](#)

## 63. Binary Tree Maximum Path Sum

Practice



## 64. Binary Tree Level Order Traversal

Practice

## 65. Serialize and Deserialize Binary Tree

Practice

## 66. Subtree of Another Tree

Practice



## 67. Construct Binary Tree from Preorder and Inorder Traversal

Practice

## 68. Validate Binary Search Tree

Practice

## 69. Kth Smallest Element in a BST

Practice

## 70. Lowest Common Ancestor of BST

Practice

## 71. Implement Trie (Prefix Tree)

Practice



## 72. Add and Search Word

Practice

## 73. Word Search II

Practice



# HEAPS

## 74. Merge K Sorted Lists

You are given an array of  $k$  linked-lists lists, each linked-list is sorted in ascending order.

Merge all the linked-lists into one sorted linked-list and return it.

[Practice](#)

## 75. Top K Frequent Elements

Given an integer array  $\text{nums}$  and an integer  $k$ , return the  $k$  most frequent elements. You may return the answer in any order.

[Practice](#)

## 76. Find Median from Data Stream

[Practice](#)



# WHY BOSSCODER?

 **2200+ Alumni** placed at Top Product-based companies.

 More than **136% hike** for every 2 out of 3 Working Professional.

 Average Package of **24LPA**.

The syllabus is most up-to-date and the list of problems provided covers all important topics.

**Lavanya**  
 Meta



Course is very well structured and streamlined to crack any MAANG company .

**Rahul**  
 Google



[EXPLORE MORE](#)