

# Most frequently asked DSA questions in MAAANG

## 1. Two Sum

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

**Input:** nums = [2,7,11,15], target = 9 **Output:** [0,1] **Explanation:**  
Because  $\text{nums}[0] + \text{nums}[1] == 9$ , we return [0, 1].

## 2. Product of Array Except Self

Given an integer array nums, return an array answer such that  $\text{answer}[i]$  is equal to the product of all the elements of nums except  $\text{nums}[i]$ .

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.

**Input:** nums =  
[1,2,3,4] **Output:**  
[24,12,8,6]

@coding\_knowledge

### 3. Median of Two Sorted Arrays

Given two sorted arrays `nums1` and `nums2` of size `m` and `n` respectively, return the median of the two sorted arrays.

The overall run time complexity should be  $O(\log(m+n))$ .

**Input:** `nums1 = [1,3]`, `nums2 = [2]`

**Output:** 2.00000

**Explanation:** merged array = `[1,2,3]` and the median is 2.

**PRACTICE NOW**

### 4. Search in Rotated Sorted Array

Given the array `nums` after the possible rotation and an integer `target`, return the index of the target if it is in `nums`, or `-1` if it is not in `nums`.

You must write an algorithm with  $O(\log n)$  runtime complexity.

**Input:** `nums = [4,5,6,7,0,1,2]`, `target = 0`

**Output:** 4

**Input:** `nums = [4,5,6,7,0,1,2]`, `target = 3`

**Output:** -1

**PRACTICE NOW**

## 5. Maximum Subarray

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

**Input:** `nums = [-2,1,-3,4,-1,2,1,-5,4]`

**Output:** 6

**Explanation:** The subarray `[4,-1,2,1]` has the largest sum 6.

**Input:** `nums = [1]`

**Output:** 1

**Explanation:** The subarray `[1]` has the largest sum of 1.

**PRACTICE NOW**

## 6. Best Time to Buy and Sell Stock

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

**Input:** `prices = [7,1,5,3,6,4]`

**Output:** 5

**Explanation:** Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = 6-1 = 5.

**PRACTICE NOW**

## 7. Trapping Rain Water

Given  $n$  non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it can trap after rain.

**Input:** height = [0,1,0,2,1,0,1,3,2,1,2,1]

**Output:** 6

**Explanation:** The above elevation map (black section) is represented by an array [0,1,0,2,1,0,1,3,2,1,2,1]. In this case, 6 units of rain water (blue section) are trapped.



**PRACTICE NOW**

@coding\_knowledge

## 8. 3Sum

Given an integer array nums, return all the triplets [nums[i], nums[j], nums[k]]

such that  $i \neq j$ ,  $i \neq k$ , and  $j \neq k$ , and  $\text{nums}[i] + \text{nums}[j] + \text{nums}[k] == 0$ .

**Notice that the solution set must not contain duplicate triplets.**

**Input:** nums = [-1,0,1,2,-1,-4]

**Output:** [[-1,-1,2],[-1,0,1]]

**Explanation:**  $\text{nums}[0] + \text{nums}[1] + \text{nums}[2] = (-1) + 0 + 1 = 0$ .

$\text{nums}[1] + \text{nums}[2] + \text{nums}[4] = 0 + 1 + (-1) = 0$ .

$\text{nums}[0] + \text{nums}[3] + \text{nums}[4] = (-1) + 2 + (-1) = 0$ .

The distinct triplets are [-1,0,1] and [-1,-1,2].

**PRACTICE NOW**

## 9. Permutations

Given an array of nums of distinct integers, return all the possible permutations. You can return the answer in any order.

**Input:** nums = [1,2,3]

**Output:** [[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]]

**PRACTICE NOW**

@coding\_knowledge

## 10. Word Search

Given an m x n grid of characters board and a string word, return true if the word exists in the grid.

The word can be constructed from letters of sequentially adjacent cells, where adjacent cells are horizontally or vertically neighboring. The same letter cell may not be used more than once.

**Input:** board = [ ["A","B","C","E"], ["S","F","C","S"], ["A","D","E","E"] ],  
word = "ABCCED"

**Output:** true

A	B	C	E
S	F	C	S
A	D	E	E

**PRACTICE NOW**

## 11. Longest Substring Without Repeating Characters

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

**Input:** s = "abccabcb"

**Output:** 3

**Explanation:** The answer is "abc", with a length of 3.

**PRACTICE NOW**

## 12. Longest Common Prefix

**Write a function to find the longest common prefix string amongst an array of strings.**

**If there is no common prefix, return an empty string "".**

**Input:** strs = ["dog","racecar","car"]

**Output:** ""

**Explanation:** There is no common prefix among the input strings.

**PRACTICE NOW**

## 13. Longest Palindromic Substring

Given a string *s*, return the longest palindromic substring in *s*.

**Input:** *s* = "babad"

**Output:** "bab"

**Explanation:** "aba" is also a valid answer.

**Input:** *s* = "cbbd"

**Output:** "bb"

@coding\_knowledge

**PRACTICE NOW**

## 14. Valid Parentheses

Given a string *s* containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

**An input string is valid if:**

1. Open brackets must be closed by the same type of brackets.
2. Open brackets must be closed in the correct order.
3. Every close bracket has a corresponding open bracket of the same type.

**Input:** *s* = "()"

**Output:** true

**Input:** *s* = "()[]{}"

**Output:** true

**Complete PDF version of this post is available in our telegram channel** 💪

**link in bio** 🔥

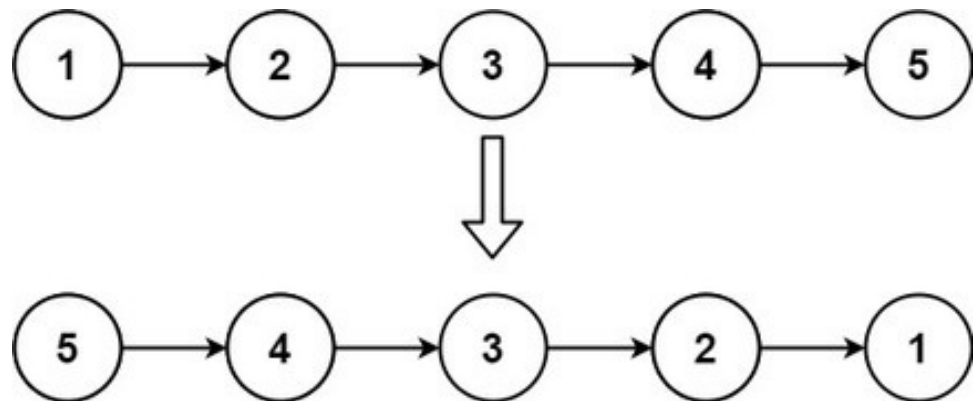
**PRACTICE NOW**

## 15. Reverse Linked Lists

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

**Input:** head = [1,2,3,4,5]

**Output:** [5,4,3,2,1]



**PRACTICE NOW**

@coding\_knowledge

## 16. Linked List Cycle

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

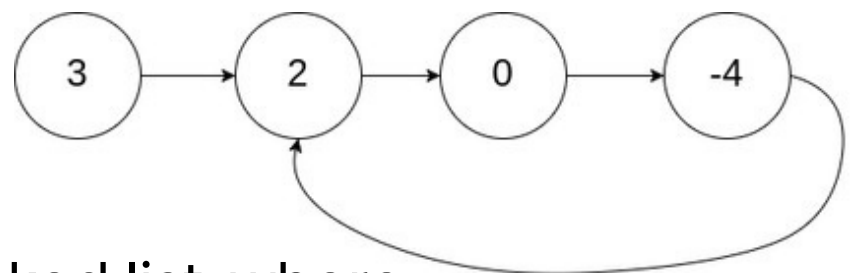
There is a cycle in a linked list if there is some node in the list that can be reached again by continuously following the next pointer. Internally, pos is used to denote the index of the node that the tail's next pointer is connected to. Note that pos is not passed as a parameter.

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

**Input:** head = [3,2,0,-4], pos = 1

**Output:** true

**Explanation:** There is a cycle in the linked list, where the tail connects to the 1st node (0-indexed).



**PRACTICE NOW**



## 17. Add Two Numbers

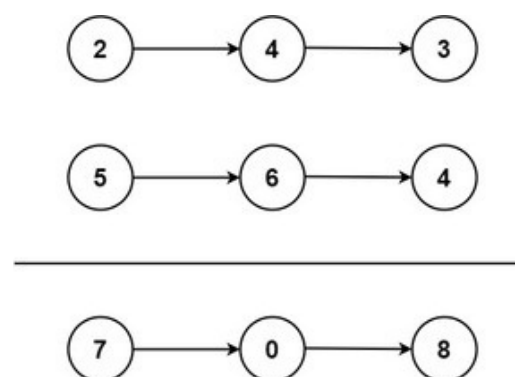
You are given two non-empty linked lists representing two non-negative integers. The digits are stored in reverse order, and each of their nodes contains a single digit. Add the two numbers and return the sum as a linked list.

You may assume the two numbers do not contain any leading zero, except the number 0 itself.

**Input:**  $l1 = [2,4,3]$ ,  $l2 = [5,6,4]$

**Output:**  $[7,0,8]$

**Explanation:**  $342 + 465 = 807$ .



**PRACTICE NOW**

## 18. 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.

**Input:**  $lists = [[1,4,5],[1,3,4],[2,6]]$

**Output:**  $[1,1,2,3,4,4,5,6]$

**Explanation:** The linked lists are:

[  
1-→4-→5,  
1-→3-→4,  
2-→6 ]

merging them into one sorted list: 1-→1-→2-→3-→4-→4-→5-→6

@coding\_knowledge

**PRACTICE NOW**

## 19. 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.**

**Input:** lists = [[1,4,5],[1,3,4],[2,6]]

**Output:** [1,1,2,3,4,4,5,6]

**Explanation:** The linked lists are:

[

1->4->5,

1->3->4,

2->6 ]

merging them into one sorted list: 1->1->2->3->4->4->5->6

**PRACTICE NOW**

## 20. Top K Frequent Elements

**Given an integer array nums and an integer k, return the k most frequent elements.**

**You may return the answer in any order.**

**Input:** nums = [1,1,1,2,2,3], k = 2

**Output:** [1,2]

**PRACTICE NOW**

**Complete PDF version of this post is available in our telegram channel  link in bio **