AlgoTutor

25 MOST FREQUENTLY ASKED DSA QUESTIONS IN MAANG

+91-7260058093

info@algotutor.io

www.algotutor.io

AlgoTutor

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 nums[0] + nums[1] == 9, we return [0, 1].

PRACTICE NOW

2. 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.

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

PRACTICE NOW

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 ith 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

AlgoTutor

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

Why Choose AlgoTutor?

- 100 % PLACEMENT ASSISTANCE
- 1:1 PERSONAL MENTORSHIP FROM INDUSTRY EXPERTS
- 100 % SUCCESS RATE
- 23 LPA(AVG.)CTC
- 200+ SUCCESSFUL ALUMNI
- 147%(AVG.) SALARY HIKE
- LEARN FROM SCRATCH
- CAREER SERVICES

8. 3Sum

Given an integer array nums, return all the triplets [nums[i], nums[j], nums[k]]
such that i != j, i != k, and j != k, and nums[i] + nums[j] + 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:
nums[0] + nums[1] + nums[2] = (-1) + 0 + 1 = 0.
nums[1] + nums[2] + nums[4] = 0 + 1 + (-1) = 0.
nums[0] + nums[3] + 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

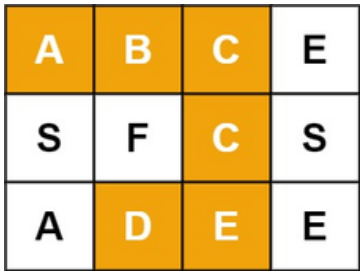
10. Word Search

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

PRACTICE NOW



11. Longest Substring Without Repeating Characters

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

Input: s = "abcabcbb"
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"

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.

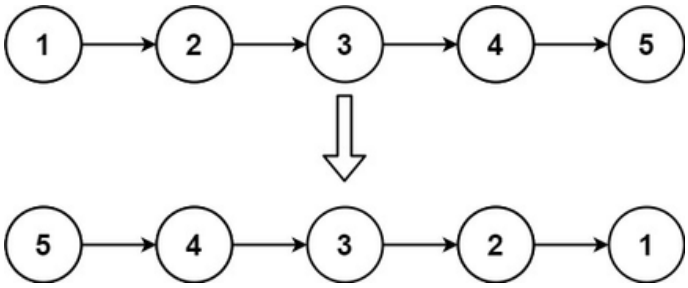
Input: s = "()"
Output: true

Input: s = "()[]{}"
Output: true

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]

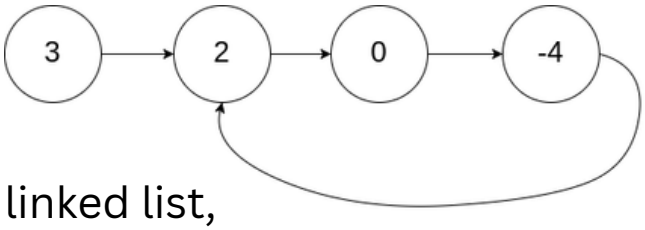
PRACTICE NOW

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).

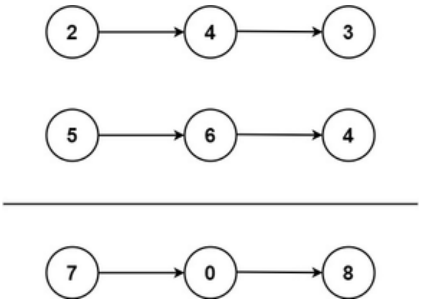
PRACTICE NOW

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

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

21. Course Schedule

There are a total of numCourses courses you have to take, labeled from 0 to numCourses - 1. You are given an array of 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.

Input: numCourses = 2, prerequisites = [[1,0]]
Output: true
Explanation: There are a total of 2 courses to take.
To take course 1 you should have finished course 0. So it is possible.

PRACTICE NOW

22. Number of Islands

Given an m x n 2D binary grid which represents a map of '1's (land) and '0's (water), return the number of islands.

An island is surrounded by water and is formed by connecting adjacent lands horizontally or vertically.

Input: grid = [["1","1","1","1","0"],
["1","1","0","1","0"],
["1","1","0","0","0"],
["0","0","0","0","0"]]
Output: 1

PRACTICE NOW

23. Non-overlapping Intervals

Given an array of intervals where intervals[i] = [starti, endi], return the minimum number of intervals you need to remove to make the rest of the intervals non-overlapping.

Input: intervals = [[1,2],[2,3],[3,4],[1,3]]
Output: 1
Explanation: [1,3] can be removed and the rest of the intervals are non-overlapping.

PRACTICE NOW

24. Meeting Rooms

Given an array of meeting time intervals consisting of start and end times[[s1,e1],[s2,e2],...](si< ei), determine if a person could attend all meetings.

Input: [[0,30],[5,10],[15,20]]
Output: false

Input: [[7,10],[2,4]]
Output: true

PRACTICE NOW

25. LRU Cache

Design a data structure that follows the constraints of a Least Recently Used (LRU) cache.
Implement the LRUCache class:

- LRUCache(int capacity) Initialize the LRU cache with positive size capacity.
- int get(int key) Return the value of the key if the key exists, otherwise return -1.
- void put(int key, int value) Update the value of the key if the key exists. Otherwise, add the key-value pair to the cache. If the number of keys exceeds the capacity from this operation, evict the least recently used key.

The functions get and put must each run in O(1) average time complexity.

Input:
["LRUCache", "put", "put", "get", "put", "get", "put", "get", "get", "get"]
[[2], [1, 1], [2, 2], [1], [3, 3], [2], [4, 4], [1], [3], [4]]
Output: [null, null, null, 1, null, -1, null, -1, 3, 4]

Explanation:
LRUCache lRUCache = new LRUCache(2);
lRUCache.put(1, 1); // cache is {1=1}
lRUCache.put(2, 2); // cache is {1=1, 2=2}
lRUCache.get(1); // return 1
lRUCache.put(3, 3); // LRU key was 2, evicts key 2, cache is {1=1, 3=3}
lRUCache.get(2); // returns -1 (not found)
lRUCache.put(4, 4); // LRU key was 1, evicts key 1, cache is {4=4, 3=3}
lRUCache.get(1); // return -1 (not found)
lRUCache.get(3); // return 3
lRUCache.get(4); // return 4

PRACTICE NOW

AlgoTutor

Why Choose AlgoTutor?

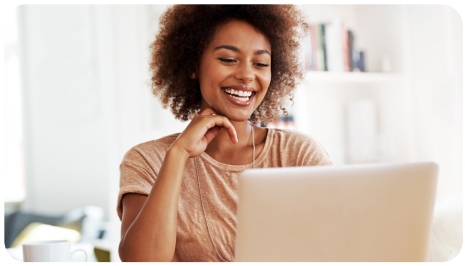
- 100 % PLACEMENT ASSISTANCE
- 1:1 PERSONAL MENTORSHIP FROM INDUSTRY EXPERTS
- 100 % SUCCESS RATE
- 23 LPA(AVG.)CTC
- 200+ SUCCESSFUL ALUMNI
- 147%(AVG.) SALARY HIKE
- LEARN FROM SCRATCH
- CAREER SERVICES

EXPLORE MORE

+91-7260058093 www.algotutor.io info@algotutor.io

AlgoTutor

Want to up your Skill?
Join our Popular courses



DSA WITH SYSTEM DESIGN (HLD + LLD)

DSA & SYSTEM DESIGN WITH FULL STACK

ADVANCED DATA SCIENCE & MACHINE LEARNING

MERN FULL STACK WEB DEVELOPMENT

EXPLORE MORE

+91-7260058093 www.algotutor.io info@algotutor.io