

# 50 QUESTIONS TO GET THROUGH MICROSOFT INTERVIEWS



## Question 1

## Next Permutation

Medium

Given an array of integers `nums`, find the next permutation of `nums`.

The next permutation of an array of integers is the next lexicographically greater permutation of its integer. If such arrangement is not possible, the array must be rearranged as the lowest possible order (i.e., sorted in ascending order).

Example 1:

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

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

Example 2:

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

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

[Problem link](#)



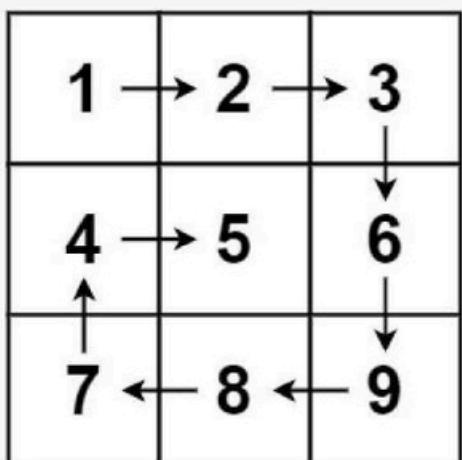
## Question 2

### Spiral Matrix

Medium

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

Example 1:



**Input:** matrix = [[1,2,3],[4,5,6],[7,8,9]]

**Output:** [1,2,3,6,9,8,7,4,5]

[Problem link](#)



## Question 3

### First Missing Positive

Hard

Given an unsorted integer array `nums`, return the smallest missing positive integer.

You must implement an algorithm that runs in  $O(n)$  time and uses  $O(1)$  auxiliary space.

**Example 1:**

**Input:** `nums` = [1,2,0]

**Output:** 3

**Explanation:** The numbers in the range [1,2] are all in the array.

**Example 2:**

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

**Output:** 2

[Problem link](#)



## Question 4

## Divide Two Integers

Medium

Given two integers dividend and divisor, divide two integers without using multiplication, division, and mod operator. The integer division should truncate toward zero, which means losing its fractional part.

Example:

**Input:** dividend = 10, divisor = 3

**Output:** 3

**Explanation:**  $10/3 = 3.33333..$  which is truncated to 3.

[Problem link](#)



## Question 5

### Pow(x,n)

Medium

Implement `pow(x, n)`, which calculates  $x$  raised to the power  $n$  (i.e.,  $x^n$ ).

Example 1:

**Input:**  $x = 2.10000$ ,  $n = 3$

**Output:** 9.26100

Example 2:

**Input:**  $x = 2.00000$ ,  $n = -2$

**Output:** 0.25000

**Explanation:**  $2^{-2} = 1/2^2 = 1/4 = 0.25$

[Problem link](#)



## Question 6

## Zigzag Conversion

Medium

The string "PAYPALISHIRING" is written in a zigzag pattern on a given number of rows like this: (you may want to display this pattern in a fixed font for better legibility)

P A H N  
A P L S I I G  
Y I R

And then read line by line: "PAHNAPLSIIGYIR"

Write the code that will take a string and make this conversion given a number of rows.

Example:

**Input:** s = "PAYPALISHIRING", numRows = 3

**Output:** "PAHNAPLSIIGYIR"

[Problem link](#)



## Question 7

### Group Anagrams

Medium

Given an array of strings `strs`, group the anagrams together. You can return the answer in any order.

An Anagram is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.

Example:

**Input:** `strs = ["eat","tea","tan","ate","nat","bat"]`

**Output:** `[["bat"], ["nat", "tan"], ["ate", "eat", "tea"]]`

[Problem link](#)



## Question 8

### String to Integer (atoi)

Medium

Implement the myAtoi(string s) function which converts a string to a 32-bit signed integer (similar to C/C++'s atoi function). Only the space character ' ' is considered a whitespace character. Do not ignore any characters other than the leading whitespace or the rest of the string after the digits.

Example 1:

**Input:** s = "42"

**Output:** 42

Example 2:

**Input:** s = " -42"

**Output:** -42

**Problem link**



## Question 9

## Search in Rotated Sorted Array

Medium

There is an integer array `nums` sorted in ascending order (with distinct values). Prior to being passed to your function, `nums` is possibly rotated at an unknown pivot index `k` ( $1 \leq k < \text{nums.length}$ ).

Given the array `nums` after the possible rotation and an integer target, return the index of 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.

## Example 1:

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

**Output:** 4

## Example 2:

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

**Output:** -1

[Problem link](#)



## Question 10

### Search a 2D Matrix

Medium

You are given an  $m \times n$  integer matrix matrix with the following two properties:

1. Each row is sorted in non-decreasing order.
2. The first integer of each row is greater than the last integer of the previous row.

Given an integer target, return true if target is in matrix or false otherwise. You must write a solution in  $O(\log(m * n))$  time complexity.

Example:

1	3	5	7
10	11	16	20
23	30	34	60

**Input:** matrix = [[1,3,5,7],[10,11,16,20],[23,30,34,60]], target = 3

**Output:** true

[Problem link](#)



## Question 11

### Median of Two Sorted Arrays

Hard

Given two sorted arrays  $\text{nums1}$  and  $\text{nums2}$  of size  $m$  and  $n$  respectively, return the median of the two sorted arrays. The overall run time complexity should be  $O(\log(m+n))$ .

Example 1:

**Input:**  $\text{nums1} = [1,3]$ ,  $\text{nums2} = [2]$

**Output:** 2.00000

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

Example 2:

**Input:**  $\text{nums1} = [1,2]$ ,  $\text{nums2} = [3,4]$

**Output:** 2.50000

**Explanation:** merged array = [1,2,3,4] and median is  $(2 + 3) / 2 = 2.5$ .

[Problem link](#)

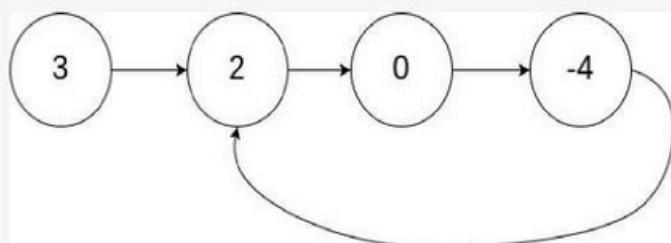


## Question 12

Linked List Cycle II Medium

Given the head of a linked list, return the node where the cycle begins. If there is no cycle, return null. Do not modify the linked list.

Example:



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

**Output:** tail connects to node index 1

**Explanation:** There is a cycle in the linked list, where tail connects to the second node.

**Problem link**



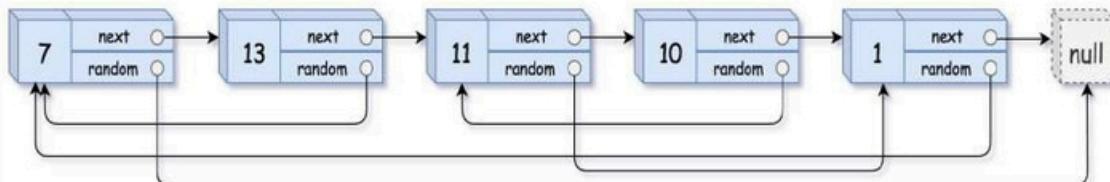
## Question 13

### Copy List with Random Pointer Medium

A linked list of length  $n$  is given such that each node  
Construct a deep copy of the list. The deep copy should consist  
of exactly  $n$  brand new nodes, where each new node has its value  
set to the value of its corresponding original node. Both the next  
and random pointer of the new nodes should point to new nodes  
in the copied list

Return the head of the copied linked list.

Example:



**Input:** head = [[7,null],[13,0],[11,4],[10,2],[1,0]]

**Output:** [[7,null],[13,0],[11,4],[10,2],[1,0]]

[Problem link](#)



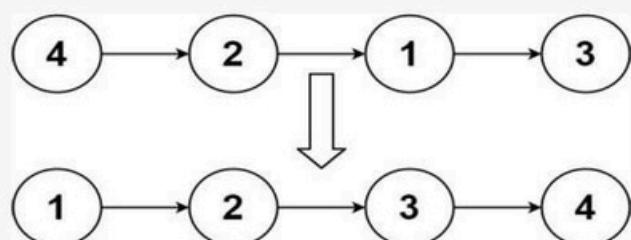
## Question 14

### Sort List

Medium

Given the head of a linked list, return the list after sorting it in ascending order.

Example:



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

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

[Problem link](#)



## Question 15

### Reverse Nodes in k-Group

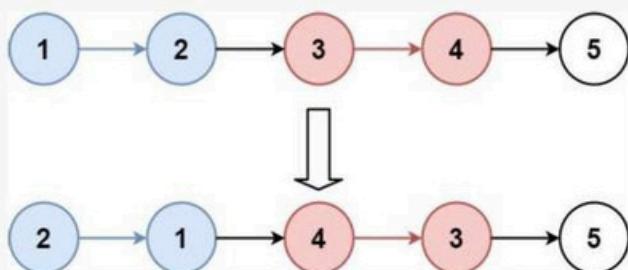
Hard

Given the head of a linked list, reverse the nodes of the list  $k$  at a time, and return the modified list.

$k$  is a positive integer and is less than or equal to the length of the linked list. If the number of nodes is not a multiple of  $k$  then left-out nodes, in the end, should remain as it is.

You may not alter the values in the list's nodes, only nodes themselves may be changed.

Example:



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

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

[Problem link](#)



## Question 16

### Min Stack

Medium

Design a stack that supports push, pop, top, and retrieving the minimum element in constant time.

Implement the MinStack class:

1. `MinStack()` initializes the stack object.
2. `void push(int val)` pushes the element `val` onto the stack.
3. `void pop()` removes the element on the top of the stack.
4. `int top()` gets the top element of the stack.
5. `int getMin()` retrieves the minimum element in the stack.

You must implement a solution with  $O(1)$  time complexity for each function.

**Example:**

**Input:**

```
["MinStack","push","push","push","getMin","pop","top","getMin"]
[],[-2],[0],[-3],[],[],[],[]]
```

**Output:**

```
[null,null,null,null,-3,null,0,-2]
```

**Problem link**



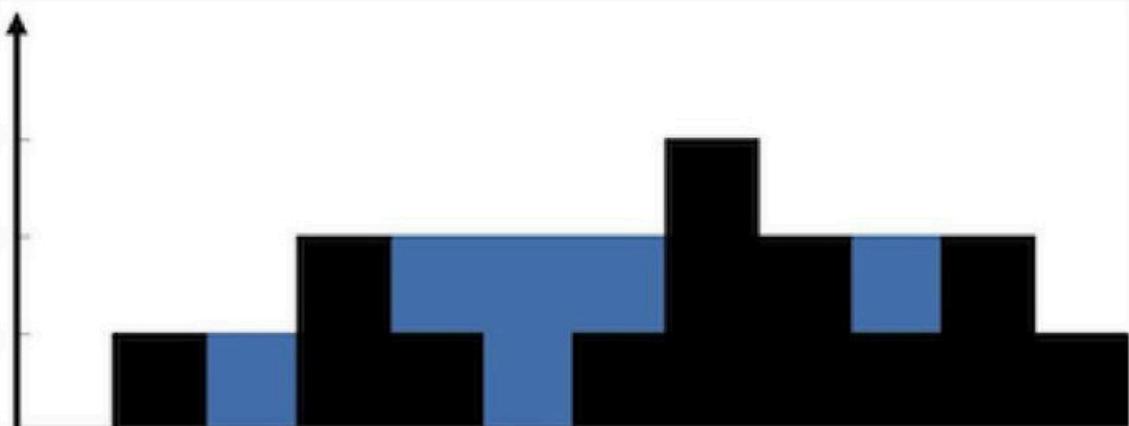
## Question 17

### Trapping Rain Water

Hard

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 raining.

Example:



**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 array [0,1,0,2,1,0,1,3,2,1,2,1]. In this case, 6 units of rain water (blue section) are being trapped.

[Problem link](#)



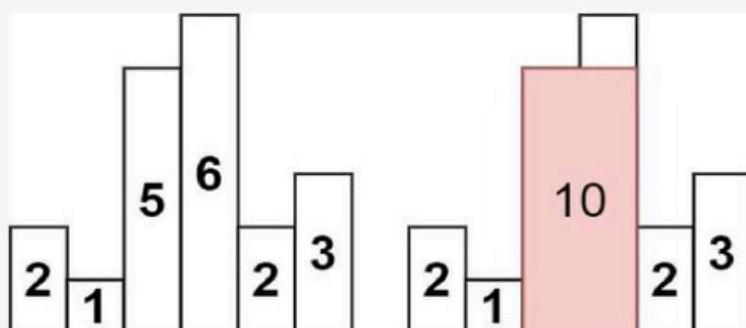
## Question 18

# Largest Rectangle in Histogram

Hard

Given an array of integers heights representing the histogram's bar height where the width of each bar is 1, return the area of the largest rectangle in the histogram.

Example:



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

**Output:** 10

**Explanation:** The above is a histogram where width of each bar is 1.

The largest rectangle is shown in the red area, which has an area = 10 units.

**Problem link**



## Question 19

### Sliding Window Maximum

Hard

You are given an array of integers `nums`, there is a sliding window of size `k` which is moving from the very left of the array to the very right. You can only see the `k` numbers in the window. Each time the sliding window moves right by one position.

Return the max sliding window.

Example:

**Input:** `nums = [1,3,-1,-3,5,3,6,7]`, `k = 3`

**Output:** `[3,3,5,5,6,7]`

Explanation:

Window position	Max
-----	-----
<code>[1 3 -1] -3 5 3 6 7</code>	<code>3</code>
<code>1 [3 -1 -3] 5 3 6 7</code>	<code>3</code>
<code>1 3 [-1 -3 5] 3 6 7</code>	<code>5</code>
<code>1 3 -1 [-3 5 3] 6 7</code>	<code>5</code>
<code>1. 3 -1 -3 [5 3 6] 7</code>	<code>6</code>
<code>1 3 -1 -3 5 [3 6 7]</code>	<code>7</code>

[Problem link](#)



## Question 20

## Word Search

Medium

Given an  $m \times n$  grid of characters board and a string word, return true if 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.

Example:

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

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

**Output:** true

[Problem link](#)



## Question 21

### N-Queens

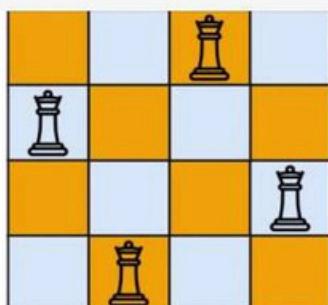
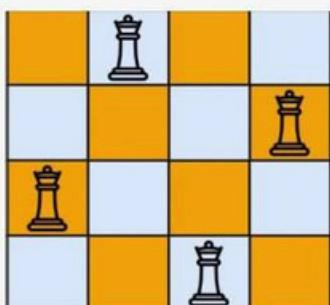
Hard

The n-queens puzzle is the problem of placing n queens on an  $n \times n$  chessboard such that no two queens attack each other.

Given an integer n, return all distinct solutions to the n-queens puzzle. You may return the answer in any order.

Each solution contains a distinct board configuration of the n-queens' placement, where 'Q' and '.' both indicate a queen and an empty space, respectively.

Example:



**Input:** n = 4

**Output:** `[[".Q..","...Q","Q...","..Q."],[..Q.","Q...","...Q",".Q.."]]`

**Explanation:** There exist two distinct solutions to the 4-queens puzzle as shown above.

[Problem link](#)



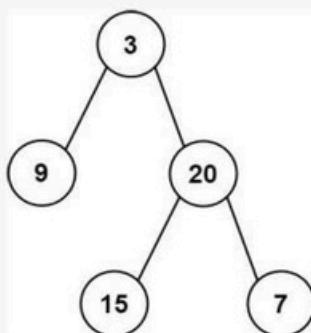
## Question 22

## Construct Binary Tree from Preorder and Inorder Traversal

Medium

Given two integer arrays preorder and inorder where preorder is the preorder traversal of a binary tree and inorder is the inorder traversal of the same tree, construct and return the binary tree.

Example:



**Input:** preorder = [3,9,20,15,7], inorder = [9,3,15,20,7]

**Output:** [3,9,20,null,null,15,7]

[Problem link](#)



## Question 23

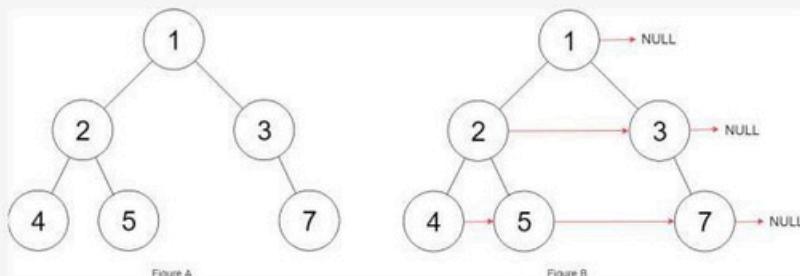
# Populating Next Right Pointers in Each Node

Medium

Given a binary tree, populate each next pointer to point to its next right node. If there is no next right node, the next pointer should be set to NULL.

Initially, all next pointers are set to NULL.

Example:



**Input:** root = [1,2,3,4,5,null,7]

**Output:** [1,#,2,3,#,4,5,7,#]

**Explanation:** Given the above binary tree (Figure A), your function should populate each next pointer to point to its next right node, just like in Figure B. The serialized output is in level order as connected by the next pointers, with '#' signifying the end of each level.

[Problem link](#)



## Question 24

### Flatten Binary Tree to Linked List

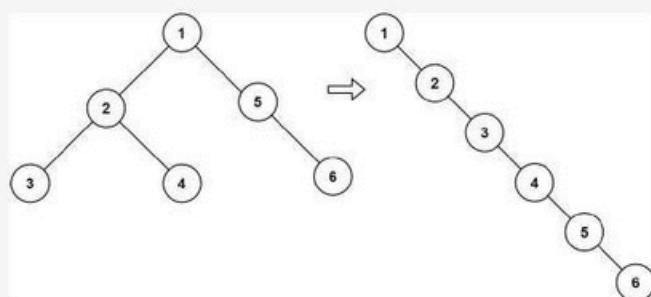
Medium

Given the root of a binary tree, flatten the tree into a "linked list":

The "linked list" should use the same `TreeNode` class where the right child pointer points to the next node in the list and the left child pointer is always null.

The "linked list" should be in the same order as a pre-order traversal of the binary tree.

Example:



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

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

[Problem link](#)

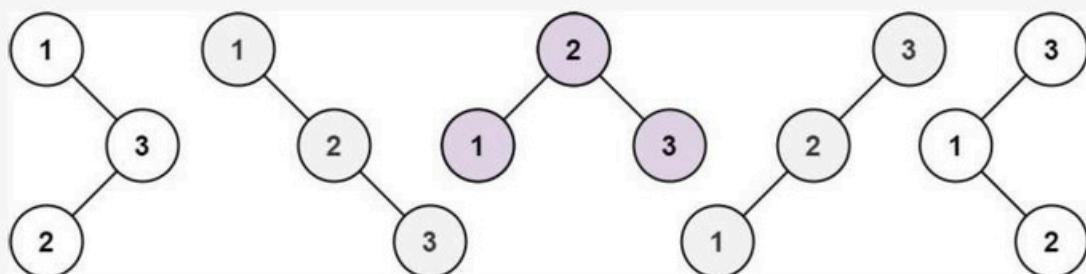


## Question 25

### Unique Binary Search Trees Medium

Given an integer n, return the number of structurally unique BST's (binary search trees) which has exactly n nodes of unique values from 1 to n.

**Example:**



**Input:** n = 3

**Output:** 5

**Problem link**



## Question 26

### Binary Search Tree Iterator

Medium

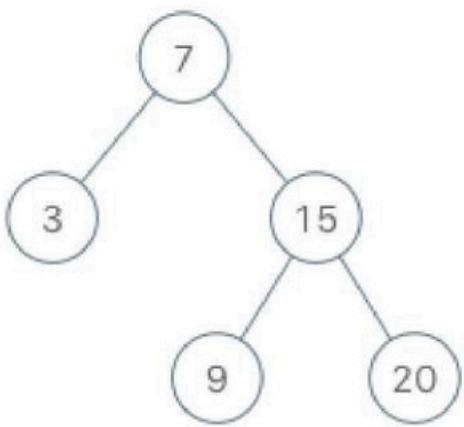
Implement the BSTIterator class that represents an iterator over the in-order traversal of a binary search tree (BST):

1. `BSTIterator(TreeNode root)` Initializes an object of the BSTIterator class. The root of the BST is given as part of the constructor. The pointer should be initialized to a non-existent number smaller than any element in the BST.
2. `boolean hasNext()` Returns true if there exists a number in the traversal to the right of the pointer, otherwise returns false.
3. `int next()` Moves the pointer to the right, then returns the number at the pointer.

You may assume that `next()` calls will always be valid.



## Example:



## Input:

```
["BSTIterator", "next", "next", "hasNext", "next", "hasNext", "next",
 "hasNext", "next", "hasNext"]
[[[7, 3, 15, null, null, 9, 20]], [], [], [], [], [], [], [], []]
```

## Output:

```
[null, 3, 7, true, 9, true, 15, true, 20, false]
```

[Problem link](#)



## Question 27

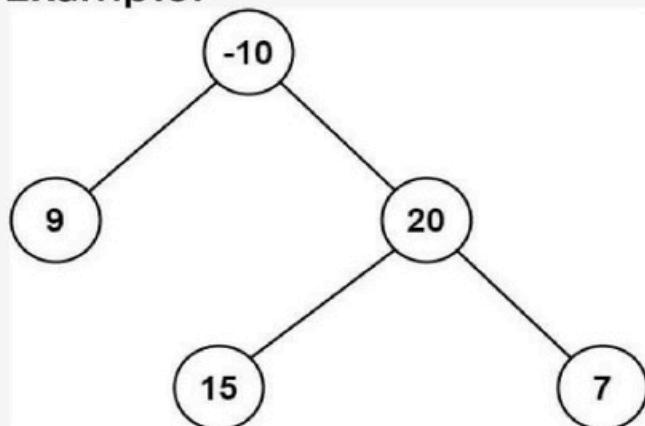
### Binary Tree Maximum Path Sum

Hard

A path in a binary tree is a sequence of nodes where each pair of adjacent nodes in the sequence has an edge connecting them. A node can only appear in the sequence at most once. Note that the path does not need to pass through the root.

Given the root of a binary tree, return the maximum path sum of any non-empty path.

Example:



**Input:** root = [-10,9,20,null,null,15,7]

**Output:** 42

**Explanation:** The optimal path is 15 -> 20 -> 7 with a path sum of  $15 + 20 + 7 = 42$ .

**Problem link**

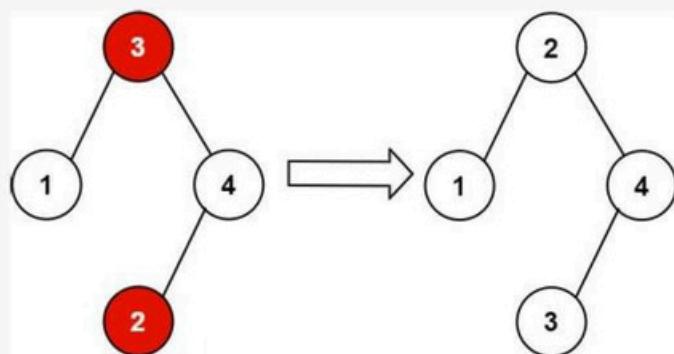


## Question 28

### Recover Binary Search Tree Hard

You are given the root of a binary search tree (BST), where the values of exactly two nodes of the tree were swapped by mistake. Recover the tree without changing its structure.

Example:



**Input:** root = [3,1,4,null,null,2]

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

**Explanation:** 2 cannot be in the right subtree of 3 because  $2 < 3$ . Swapping 2 and 3 makes the BST valid.



## Question 29

## Longest Substring Without Repeating Characters

Medium

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

**Example:**

**Input:** s = "abcabcbb"

**Output:** 3

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

[Problem link](#)



## Question 30

### Insert Delete GetRandom O(1)

Medium

Implement the RandomizedSet class:

1. `RandomizedSet()` Initializes the RandomizedSet object.
2. `bool insert(int val)` Inserts an item `val` into the set if not present. Returns true if the item was not present, false otherwise.
3. `bool remove(int val)` Removes an item `val` from the set if present. Returns true if the item was present, false otherwise.
4. `int getRandom()` Returns a random element from the current set of elements (it's guaranteed that at least one element exists when this method is called). Each element must have the same probability of being returned.

Each function should work in average O(1) time complexity.

**Example:**

**Input:**

```
["RandomizedSet", "insert", "remove", "insert", "getRandom",
 "remove", "insert", "getRandom"]
[], [1], [2], [2], [], [1], [2], []]
```

**Output:**

```
[null, true, false, true, 2, true, false, 2]
```

**Problem link**



## Question 31

### LRU Cache

Medium

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

Implement the LRUCache class:

1. `LRUCache(int capacity)` Initialize the LRU cache with positive size capacity.
2. `int get(int key)` Return the value of the key if the key exists, otherwise return -1.
3. `void put(int key, int value)` Update the value of the key if the key exists.

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

**Example:**

**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]
```

**Problem link**



## Question 32

### Minimum Window Substring

Hard

Given two strings  $s$  and  $t$  of lengths  $m$  and  $n$  respectively, return the minimum window substring of  $s$  such that every character in  $t$  (including duplicates) is included in the window. If there is no such substring, return the empty string  $""$ . The testcases will be generated such that the answer is unique.

Example:

**Input:**  $s = \text{"ADOBECODEBANC"}$ ,  $t = \text{"ABC"}$

**Output:**  $\text{"BANC"}$

**Explanation:** The minimum window substring "BANC" includes 'A', 'B', and 'C' from string  $t$ .

[Problem link](#)



## Question 33

## Find Median from Data Stream

Hard

The median is the middle value in an ordered integer list. If the size of the list is even, there is no middle value, and the median is the mean of the two middle values.

For example, for arr = [2,3,4], the median is 3.

For example, for arr = [2,3], the median is  $(2 + 3) / 2 = 2.5$ .

Implement the MedianFinder class:

1. MedianFinder() initializes the MedianFinder object.
2. void addNum(int num) adds the integer num from the data stream to the data structure.
3. double findMedian() returns the median of all elements so far. Answers within 10-5 of the actual answer will be accepted.

Example:

**Input:**

```
["MedianFinder", "addNum", "addNum", "findMedian", "addNum",
 "findMedian"]
```

```
[[], [1], [2], [], [3], []]
```

**Output:**

```
[null, null, null, 1.5, null, 2.0]
```

**Problem link**



## Question 34

## Merge Intervals

Medium

Given an array of intervals where  $\text{intervals}[i] = [\text{start}_i, \text{end}_i]$ , merge all overlapping intervals, and return an array of the non-overlapping intervals that cover all the intervals in the input.

Example:

**Input:** intervals = [[1,3],[2,6],[8,10],[15,18]]

**Output:** [[1,6],[8,10],[15,18]]

**Explanation:** Since intervals [1,3] and [2,6] overlap, merge them into [1,6].

[Problem link](#)



## Question 35

### Jump Game

Medium

You are given an integer array `nums`. You are initially positioned at the array's first index, and each element in the array represents your maximum jump length at that position. Return true if you can reach the last index, or false otherwise.

**Example 1:**

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

**Output:** true

**Explanation:** Jump 1 step from index 0 to 1, then 3 steps to the last index.

**Example 2:**

**Input:** `nums = [3,2,1,0,4]`

**Output:** false

**Explanation:** You will always arrive at index 3 no matter what. Its maximum jump length is 0, which makes it impossible to reach the last index.

**Problem link**



## Question 36

### Gas Station

Medium

There are  $n$  gas stations along a circular route, where the amount of gas at the  $i$ th station is  $\text{gas}[i]$ . You have a car with an unlimited gas tank and it costs  $\text{cost}[i]$  of gas to travel from the  $i$ th station to its next  $(i + 1)$ th station. You begin the journey with an empty tank at one of the gas stations.

Given two integer arrays  $\text{gas}$  and  $\text{cost}$ , return the starting gas station's index if you can travel around the circuit once in the clockwise direction, otherwise return -1.

Example 1:

**Input:**  $\text{gas} = [1,2,3,4,5]$ ,  $\text{cost} = [3,4,5,1,2]$

**Output:** 3

Example 2:

**Input:**  $\text{gas} = [2,3,4]$ ,  $\text{cost} = [3,4,3]$

**Output:** -1

[Problem link](#)



## Question 37

### Candy

Hard

There are  $n$  children standing in a line. Each child is assigned a rating value given in the integer array ratings. You are giving candies to these children subjected to the following requirements:

1. Each child must have at least one candy.
2. Children with a higher rating get more candies than their neighbors.

Return the minimum number of candies you need to have to distribute the candies to the children.

Example:

**Input:** ratings = [1,0,2]

**Output:** 5

**Explanation:** You can allocate to the first, second and third child with 2, 1, 2 candies respectively.

[Problem link](#)



## Question 38

## Longest Increasing Subsequence

Medium

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

Example:

**Input:** nums = [10,9,2,5,3,7,101,18]

**Output:** 4

**Explanation:** The longest increasing subsequence is [2,3,7,101], therefore the length is 4.

[Problem link](#)



## Question 39

### Maximum Product Subarray

Medium

Given an integer array `nums`, find a subarray that has the largest product, and return the product. The test cases are generated so that the answer will fit in a 32-bit integer.

**Example 1:**

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

**Output:** 6

**Explanation:** `[2,3]` has the largest product 6.

**Example 2:**

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

**Output:** 0

**Explanation:** The result cannot be 2, because `[-2,-1]` is not a subarray.

[Problem link](#)



## Question 40

### Decode Ways

Medium

A message containing letters from A-Z can be encoded into numbers using the following mapping:

'A' -> "1", 'B' -> "2" , ..., 'Z' -> "26"

Given a string s containing only digits, return the number of ways to decode it.

**Example:**

**Input:** s = "226"

**Output:** 3

**Explanation:** "226" could be decoded as "BZ" (2 26), "VF" (22 6), or "BBF" (2 2 6).

**Problem link**



## Question 41

### Word Break

Medium

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.

Example:

**Input:** s = "leetcode", wordDict = ["leet", "code"]

**Output:** true

**Explanation:** Return true because "leetcode" can be segmented as "leet code".

[Problem link](#)



## Question 42

### Interleaving String

Medium

Given strings  $s_1$ ,  $s_2$ , and  $s_3$ , find whether  $s_3$  is formed by an interleaving of  $s_1$  and  $s_2$ . An interleaving of two strings  $s$  and  $t$  is a configuration where  $s$  and  $t$  are divided into  $n$  and  $m$  substrings respectively, such that:

$$s = s_1 + s_2 + \dots + s_n$$

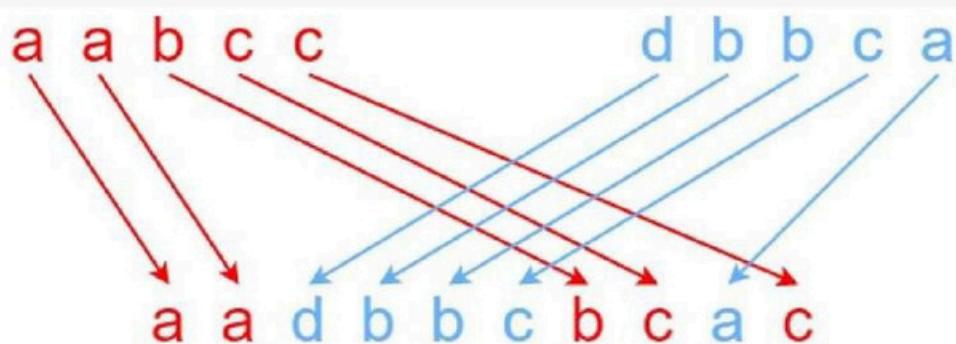
$$t = t_1 + t_2 + \dots + t_m$$

$$|n - m| \leq 1$$

The interleaving is  $s_1 + t_1 + s_2 + t_2 + s_3 + t_3 + \dots$  or  $t_1 + s_1 + t_2 + s_2 + t_3 + s_3 + \dots$

Note:  $a + b$  is the concatenation of strings  $a$  and  $b$ .

Example:



**Input:**  $s_1 = "aabcc"$ ,  $s_2 = "dbbca"$ ,  $s_3 = "aadbcbcbcac"$

**Output:** true

[Problem link](#)



## Question 43

### Edit Distance

Hard

Given two strings word1 and word2, return the minimum number of operations required to convert word1 to word2.

You have the following three operations permitted on a word:  
Insert a character, Delete a character or Replace a character.

**Example:**

**Input:** word1 = "horse", word2 = "ros"

**Output:** 3

**Explanation:**

horse -> rorse (replace 'h' with 'r')

rorse -> rose (remove 'r')

rose -> ros (remove 'e')

**Problem link**



## Question 44

### Distinct Subsequences

Hard

Given two strings s and t, return the number of distinct subsequences of s which equals t.

The test cases are generated so that the answer fits on a 32-bit signed integer.

**Example:**

**Input:** s = "rabbbit", t = "rabbit"

**Output:** 3

**Explanation:**

As shown below, there are 3 ways you can generate "rabbit" from s.

[Problem link](#)



## Question 45

### Wildcard Matching

Hard

Given an input string ( $s$ ) and a pattern ( $p$ ), implement wildcard pattern matching with support for '?' and '\*' where:

'?' Matches any single character.

"\*" Matches any sequence of characters (including the empty sequence).

The matching should cover the entire input string (not partial).

Example 1:

**Input:**  $s = "aa"$ ,  $p = "*"$

**Output:** true

**Explanation:** '\*' matches any sequence.

Example 2:

**Input:**  $s = "cb"$ ,  $p = "?a"$

**Output:** false

**Explanation:** '?' matches 'c', but the second letter is 'a', which does not match 'b'.

[Problem link](#)



## Question 46

### Best Time to Buy and Sell Stock III

Hard

You are given an array `prices` where `prices[i]` is the price of a given stock on the  $i$ th day. Find the maximum profit you can achieve. You may complete at most two transactions.

Note: You may not engage in multiple transactions simultaneously (i.e., you must sell the stock before you buy again).

Example:

**Input:** `prices = [3,3,5,0,0,3,1,4]`

**Output:** 6

**Explanation:** Buy on day 4 (price = 0) and sell on day 6 (price = 3), profit =  $3-0 = 3$ .

Then buy on day 7 (price = 1) and sell on day 8 (price = 4), profit =  $4-1 = 3$ .

[Problem link](#)



## Question 47

### Palindrome Partitioning II Hard

Given a string  $s$ , partition  $s$  such that every substring of the partition is a palindrome. Return the minimum cuts needed for a palindrome partitioning of  $s$ .

**Example:**

**Input:**  $s = "aab"$

**Output:** 1

**Explanation:** The palindrome partitioning  $["aa", "b"]$  could be produced using 1 cut.

**Problem link**



## Question 48

## Surrounded Regions

Medium

Given an  $m \times n$  matrix board containing 'X' and 'O', capture all regions that are 4-directionally surrounded by 'X'. A region is captured by flipping all 'O's into 'X's in that surrounded region.

**Example:**

X	X	X	X
X	O	O	X
X	X	O	X
X	O	X	X

→

X	X	X	X
X	X	X	X
X	X	X	X
X	O	X	X

**Input:** board = [["X","X","X","X"],["X","O","O","X"],["X","X","O","X"], ["X","O","X","X"]]

**Output:** [[“X”, “X”, “X”, “X”], [“X”, “X”, “X”, “X”], [“X”, “X”, “X”, “X”], [“X”, “O”, “X”, “X”]]

**Explanation:** Notice that an 'O' should not be flipped if:

- It is on the border, or
- It is adjacent to an 'O' that should not be flipped.

The bottom 'O' is on the border, so it is not flipped.

The other three 'O' form a surrounded region, so they are flipped.

**Problem link**



## Question 49

### Course Schedule

Medium

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.

**Example 1:**

**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.

**Example 2:**

**Input:** `numCourses = 2, prerequisites = [[1,0],[0,1]]`

**Output:** false

**Explanation:** There are a total of 2 courses to take.

To take course 1 you should have finished course 0, and to take course 0 you should also have finished course 1. So it is impossible.

**Problem link**



## Question 50

### Word Ladder Hard

A transformation sequence from word `beginWord` to word `endWord` using a dictionary `wordList` is a sequence of words `beginWord` -> `s1` -> `s2` -> ... -> `sk` such that:

Every adjacent pair of words differs by a single letter.

Every `si` for  $1 \leq i \leq k$  is in `wordList`. Note that `beginWord` does not need to be in `wordList`.

`sk` == `endWord`

Given two words, `beginWord` and `endWord`, and a dictionary `wordList`, return the number of words in the shortest transformation sequence from `beginWord` to `endWord`, or 0 if no such sequence exists.

**Example:**

**Input:** `beginWord` = "hit", `endWord` = "cog", `wordList` = ["hot", "dot", "dog", "lot", "log", "cog"]

**Output:** 5

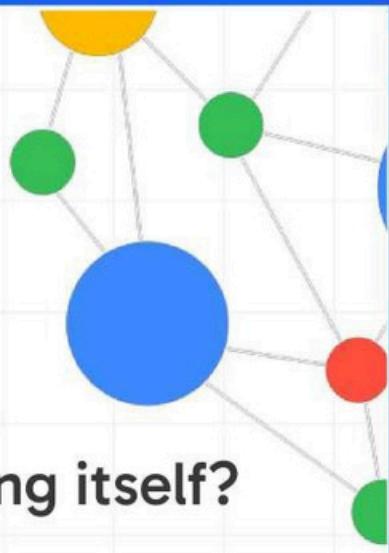
**Explanation:** One shortest transformation sequence is "hit" -> "hot" -> "dot" -> "dog" -> "cog", which is 5 words long.

**Problem link**



# Curious how to crack Microsoft?

Not getting calls for the initial screening itself?



- **3000+** engineers chose HeyCoach to land a job at a top product company.
- ↗ Already working professionals are raising their salaries by **300%** with CTC up to **1.2 CR!**

The batch size of 30 students gives the learners a great level of personalized attention



Shubham  
Deloitte

They assigned me a coach who helped me with the preparations and also guided me throughout the interview process.



Monark  
Razorpay

[Check Eligibility](#)