

134. Gas Station

Solved

Medium

Topics

Companies

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 $(i + 1)^{\text{th}}$ station. You begin the journey with an empty tank at one of the gas stations.

Given two integer arrays gas and cost , return the starting gas station's index if you can travel around the circuit once in the clockwise direction, otherwise return -1. If there exists a solution, it is guaranteed to be unique.

Example 1:

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

Output: 3

Explanation:

Start at station 3 (index 3) and fill up with 4 unit of gas. Your tank = $0 + 4 = 4$

Travel to station 4. Your tank = $4 - 1 + 5 = 8$

Travel to station 0. Your tank = $8 - 2 + 1 = 7$

Travel to station 1. Your tank = $7 - 3 + 2 = 6$

Travel to station 2. Your tank = $6 - 4 + 3 = 5$

Travel to station 3. The cost is 5. Your gas is just enough to travel back to station 3.

Therefore, return 3 as the starting index.

Example 2:

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

Output: -1

Explanation:

You can't start at station 0 or 1, as there is not enough gas to travel to the next station.

Let's start at station 2 and fill up with 4 unit of gas. Your tank = $0 + 4 = 4$

Travel to station 0. Your tank = $4 - 3 + 2 = 3$

Travel to station 1. Your tank = $3 - 3 + 3 = 3$

You cannot travel back to station 2, as it requires 4 unit of gas but you only have 3.

Therefore, you can't travel around the circuit once no matter where you start.

Constraints:

- $n == \text{gas.length} == \text{cost.length}$
 - $1 \leq n \leq 10^5$
 - $0 \leq \text{gas}[i], \text{cost}[i] \leq 10^4$
-

80. Remove Duplicates from Sorted Array II

Solved

Medium

Topics

Companies

Given an integer array **nums** sorted in **non-decreasing order**, remove some duplicates **in-place** such that each unique element appears **at most twice**. The **relative order** of the elements should be kept the **same**.

Since it is impossible to change the length of the array in some languages, you must instead have the result be placed in the **first part** of the array **nums**. More formally, if there are k elements after removing the duplicates, then the first k elements of **nums** should hold the final result. It does not matter what you leave beyond the first k elements.

Return k *after placing the final result in the first k slots of **nums**.*

Do **not** allocate extra space for another array. You must do this by **modifying the input array in-place** with $O(1)$ extra memory.

Custom Judge:

The judge will test your solution with the following code:

```
int[] nums = [...]; // Input array
int[] expectedNums = [...]; // The expected answer with correct length

int k = removeDuplicates(nums); // Calls your implementation

assert k == expectedNums.length;
for (int i = 0; i < k; i++) {
    assert nums[i] == expectedNums[i];
}
```

If all assertions pass, then your solution will be **accepted**.

Example 1:**Input:** `nums = [1,1,1,2,2,3]`**Output:** `5, nums = [1,1,2,2,3,_]`**Explanation:** Your function should return $k = 5$, with the first five elements of `nums` being 1, 1, 2, 2 and 3 respectively.It does not matter what you leave beyond the returned k (hence they are underscores).**Example 2:****Input:** `nums = [0,0,1,1,1,1,2,3,3]`**Output:** `7, nums = [0,0,1,1,2,3,3,_,_]`**Explanation:** Your function should return $k = 7$, with the first seven elements of `nums` being 0, 0, 1, 1, 2, 3 and 3 respectively.It does not matter what you leave beyond the returned k (hence they are underscores).**Constraints:**

- $1 \leq \text{nums.length} \leq 3 \cdot 10^4$
 - $-10^4 \leq \text{nums}[i] \leq 10^4$
 - `nums` is sorted in **non-decreasing** order.
-

189. Rotate Array

Solved

Medium

Topics

Companies

Hint

Given an integer array `nums`, rotate the array to the right by k steps, where k is non-negative.**Example 1:****Input:** `nums = [1,2,3,4,5,6,7], k = 3`**Output:** `[5,6,7,1,2,3,4]`**Explanation:**rotate 1 steps to the right: `[7,1,2,3,4,5,6]`rotate 2 steps to the right: `[6,7,1,2,3,4,5]`

rotate 3 steps to the right: [5,6,7,1,2,3,4]

Example 2:

Input: nums = [-1,-100,3,99], k = 2

Output: [3,99,-1,-100]

Explanation:

rotate 1 steps to the right: [99,-1,-100,3]

rotate 2 steps to the right: [3,99,-1,-100]

Constraints:

- $1 \leq \text{nums.length} \leq 10^5$
- $-2^{31} \leq \text{nums}[i] \leq 2^{31} - 1$
- $0 \leq k \leq 10^5$

122. Best Time to Buy and Sell Stock II

Solved

Medium

Topics

Companies

You are given an integer array prices where $\text{prices}[i]$ is the price of a given stock on the i^{th} day.

On each day, you may decide to buy and/or sell the stock. You can only hold **at most one** share of the stock at any time. However, you can buy it then immediately sell it on the **same day**.

Find and return the maximum profit you can achieve.

Example 1:

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

Output: 7

Explanation: Buy on day 2 (price = 1) and sell on day 3 (price = 5), profit = 5-1 = 4.

Then buy on day 4 (price = 3) and sell on day 5 (price = 6), profit = 6-3 = 3.

Total profit is 4 + 3 = 7.

Example 2:

Input: prices = [1,2,3,4,5]

Output: 4

Explanation: Buy on day 1 (price = 1) and sell on day 5 (price = 5), profit = $5 - 1 = 4$.

Total profit is 4.

Example 3:

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

Output: 0

Explanation: There is no way to make a positive profit, so we never buy the stock to achieve the maximum profit of 0.

Constraints:

- $1 \leq \text{prices.length} \leq 3 \cdot 10^4$
 - $0 \leq \text{prices}[i] \leq 10^4$
-

55. Jump Game

Solved

Medium

Topics

Companies

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.

Constraints:

- $1 \leq \text{nums.length} \leq 10^4$
 - $0 \leq \text{nums}[i] \leq 10^5$
-

Testcase

Test Result

Test Result

Code

55. Jump Game

Solved

Medium

Topics

Companies

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.

Constraints:

- $1 \leq \text{nums.length} \leq 10^4$
 - $0 \leq \text{nums}[i] \leq 10^5$
-

Description

Editorial

Editorial

Solutions

Solutions

(click)

Submissions

Submissions

Testcase

Test Result

Test Result

Code

[274. H-Index](#)

Solved

Medium

Topics

Companies

Hint

Given an array of integers citations where citations[i] is the number of citations a researcher received for their i^{th} paper, return *the researcher's h-index*.

According to the [definition of h-index on Wikipedia](#): The h-index is defined as the maximum value of h such that the given researcher has published at least h papers that have each been cited at least h times.

Example 1:

Input: citations = [3,0,6,1,5]

Output: 3

Explanation: [3,0,6,1,5] means the researcher has 5 papers in total and each of them had received 3, 0, 6, 1, 5 citations respectively.

Since the researcher has 3 papers with at least 3 citations each and the remaining two with no more than 3 citations each, their h-index is 3.

Example 2:

Input: citations = [1,3,1]

Output: 1

Constraints:

- $n == \text{citations.length}$
- $1 \leq n \leq 5000$
- $0 \leq \text{citations}[i] \leq 1000$

380. Insert Delete GetRandom O(1)

Solved

Medium

Topics

Companies

Implement the RandomizedSet class:

- RandomizedSet() Initializes the RandomizedSet object.
- `bool insert(int val)` Inserts an item val into the set if not present. Returns true if the item was not present, false otherwise.
- `bool remove(int val)` Removes an item val from the set if present. Returns true if the item was present, false otherwise.
- `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.

You must implement the functions of the class such that each function works in **average** $O(1)$ time complexity.

Example 1:

Input

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

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

Output

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

Explanation

```
RandomizedSet randomizedSet = new RandomizedSet();
```

```
randomizedSet.insert(1); // Inserts 1 to the set. Returns true as 1 was inserted successfully.
```

```
randomizedSet.remove(2); // Returns false as 2 does not exist in the set.
```



```
randomizedSet.insert(2); // Inserts 2 to the set, returns true. Set now contains [1,2].
randomizedSet.getRandom(); // getRandom() should return either 1 or 2 randomly.
randomizedSet.remove(1); // Removes 1 from the set, returns true. Set now contains [2].
randomizedSet.insert(2); // 2 was already in the set, so return false.
randomizedSet.getRandom(); // Since 2 is the only number in the set, getRandom() will always return 2.
```

Constraints:

- $-2^{31} \leq \text{val} \leq 2^{31} - 1$
- At most $2 * 10^5$ calls will be made to insert, remove, and getRandom.
- There will be **at least one** element in the data structure when getRandom is called.

238. Product of Array Except Self

Solved

Medium

Topics

Companies

Hint

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.

Example 1:

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

Output: [24,12,8,6]

Example 2:

Input: *nums* = [-1,1,0,-3,3]

Output: [0,0,9,0,0]

Constraints:

- $2 \leq \text{nums.length} \leq 10^5$

- $-30 \leq \text{nums}[i] \leq 30$
- The input is generated such that $\text{answer}[i]$ is **guaranteed** to fit in a **32-bit** integer.

12. Integer to Roman

Solved

Medium

Topics

Companies

Seven different symbols represent Roman numerals with the following values:

Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

Roman numerals are formed by appending the conversions of decimal place values from highest to lowest. Converting a decimal place value into a Roman numeral has the following rules:

- If the value does not start with 4 or 9, select the symbol of the maximal value that can be subtracted from the input, append that symbol to the result, subtract its value, and convert the remainder to a Roman numeral.
- If the value starts with 4 or 9 use the **subtractive form** representing one symbol subtracted from the following symbol, for example, 4 is 1 (I) less than 5 (V): IV and 9 is 1 (I) less than 10 (X): IX. Only the following subtractive forms are used: 4 (IV), 9 (IX), 40 (XL), 90 (XC), 400 (CD) and 900 (CM).
- Only powers of 10 (I, X, C, M) can be appended consecutively at most 3 times to represent multiples of 10. You cannot append 5 (V), 50 (L), or 500 (D) multiple times. If you need to append a symbol 4 times use the **subtractive form**.

Given an integer, convert it to a Roman numeral.

Example 1:

Input: num = 3749

Output: "MMMDCCLXIX"

Explanation:

3000 = MMM as 1000 (M) + 1000 (M) + 1000 (M)

700 = DCC as 500 (D) + 100 (C) + 100 (C)

40 = XL as 10 (X) less of 50 (L)

9 = IX as 1 (I) less of 10 (X)

Note: 49 is not 1 (I) less of 50 (L) because the conversion is based on decimal places

Example 2:

Input: num = 58

Output: "LVIII"

Explanation:

50 = L

8 = VIII

Example 3:

Input: num = 1994

Output: "MCMXCIV"

Explanation:

1000 = M

900 = CM

90 = XC

4 = IV

Constraints:

- $1 \leq \text{num} \leq 3999$
-

151. Reverse Words in a String

Solved

Medium

Topics

Companies

Given an input string *s*, reverse the order of the **words**.

A **word** is defined as a sequence of non-space characters. The **words** in *s* will be separated by at least one space.

Return *a string of the words in reverse order concatenated by a single space*.

Note that *s* may contain leading or trailing spaces or multiple spaces between two words. The returned string should only have a single space separating the words. Do not include any extra spaces.

Example 1:

Input: *s* = "the sky is blue"

Output: "blue is sky the"

Example 2:

Input: *s* = " hello world "

Output: "world hello"

Explanation: Your reversed string should not contain leading or trailing spaces.

Example 3:

Input: *s* = "a good example"

Output: "example good a"

Explanation: You need to reduce multiple spaces between two words to a single space in the reversed string.

Constraints:

- $1 \leq s.length \leq 10^4$
- *s* contains English letters (upper-case and lower-case), digits, and spaces ' '.
- There is **at least one** word in *s*.

6. Zigzag Conversion

12 Solved

Medium

Topics

Companies

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:

string convert(string s, int numRows);

Example 1:

Input: s = "PAYPALISHIRING", numRows = 3

Output: "PAHNAPLSIIGYIR"

Example 2:

Input: s = "PAYPALISHIRING", numRows = 4

Output: "PINALSIGYAHRPI"

Explanation:

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

Example 3:

Input: s = "A", numRows = 1

Output: "A"

Constraints:

- $1 \leq s.length \leq 1000$
- s consists of English letters (lower-case and upper-case), ',' and '!'.
- $1 \leq numRows \leq 1000$

167. Two Sum II - Input Array Is Sorted

Solved

Medium

Topics

13

Companies

Given a **1-indexed** array of integers numbers that is already **sorted in non-decreasing order**, find two numbers such that they add up to a specific target number. Let these two numbers be numbers[index₁] and numbers[index₂] where $1 \leq \text{index}_1 < \text{index}_2 \leq \text{numbers.length}$.

Return *the indices of the two numbers, index₁ and index₂, added by one as an integer array [index₁, index₂] of length 2.*

The tests are generated such that there is **exactly one solution**. You **may not** use the same element twice.

Your solution must use only constant extra space.

Example 1:

Input: numbers = [2,7,11,15], target = 9

Output: [1,2]

Explanation: The sum of 2 and 7 is 9. Therefore, index₁ = 1, index₂ = 2. We return [1, 2].

Example 2:

Input: numbers = [2,3,4], target = 6

Output: [1,3]

Explanation: The sum of 2 and 4 is 6. Therefore index₁ = 1, index₂ = 3. We return [1, 3].

Example 3:

Input: numbers = [-1,0], target = -1

Output: [1,2]

Explanation: The sum of -1 and 0 is -1. Therefore index₁ = 1, index₂ = 2. We return [1, 2].

Constraints:

- $2 \leq \text{numbers.length} \leq 3 * 10^4$
- $-1000 \leq \text{numbers}[i] \leq 1000$
- numbers is sorted in **non-decreasing order**.
- $-1000 \leq \text{target} \leq 1000$
- The tests are generated such that there is **exactly one solution**.

11. Container With Most Water

Solved

Medium

Topics

Companies

Hint

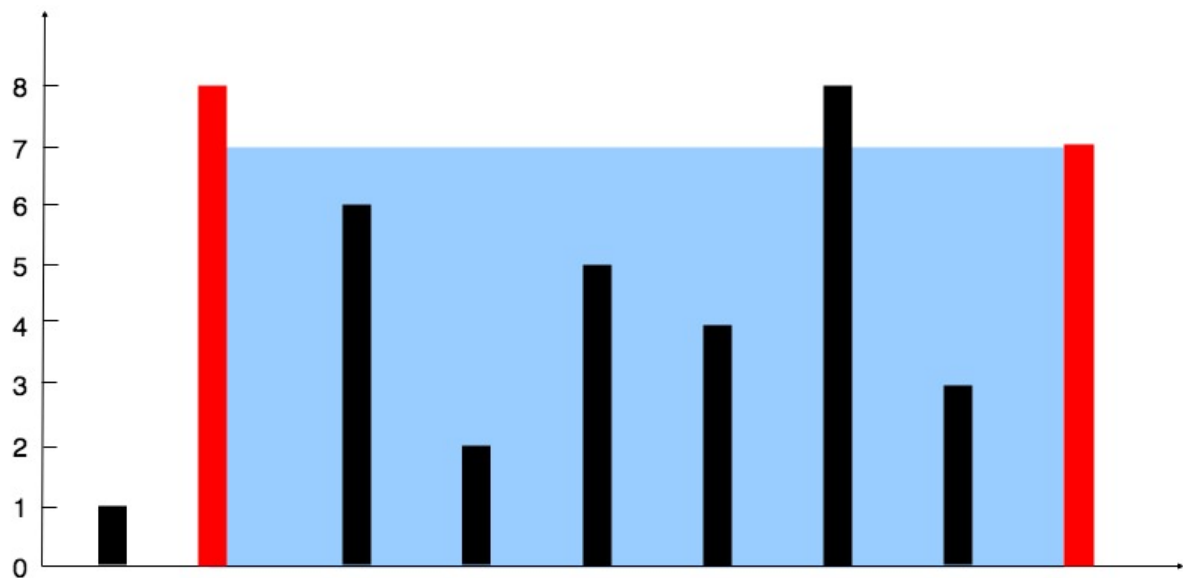
You are given an integer array `height` of length `n`. There are `n` vertical lines drawn such that the two endpoints of the i^{th} line are $(i, 0)$ and $(i, \text{height}[i])$.

Find two lines that together with the x-axis form a container, such that the container contains the most water.

Return *the maximum amount of water a container can store*.

Notice that you may not slant the container.

Example 1:



Input: `height = [1,8,6,2,5,4,8,3,7]`

Output: 49

Explanation: The above vertical lines are represented by array `[1,8,6,2,5,4,8,3,7]`. In this case, the max area of water (blue section) the container can contain is 49.

Example 2:

Input: `height = [1,1]`

Output: 1

Constraints:

- `n == height.length`

- $2 \leq n \leq 10^5$
- $0 \leq \text{height}[i] \leq 10^4$

15. 3Sum

Solved

Medium

Topics

Companies

Hint

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.

Example 1:

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]`.

Notice that the order of the output and the order of the triplets does not matter.

Example 2:

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

Output: `[]`

Explanation: The only possible triplet does not sum up to 0.

Example 3:

Input: `nums = [0,0,0]`

Output: `[[0,0,0]]`

Explanation: The only possible triplet sums up to 0.

Constraints:

- $3 \leq \text{nums.length} \leq 3000$
 - $-10^5 \leq \text{nums}[i] \leq 10^5$
-

Testcase

Test Result

Test Result

Code

[209. Minimum Size Subarray Sum](#)

Solved

Medium

Topics

Companies

Given an array of positive integers `nums` and a positive integer `target`, return *the minimal length of a subarray whose sum is greater than or equal to target*. If there is no such subarray, return 0 instead.

Example 1:

Input: `target = 7, nums = [2,3,1,2,4,3]`

Output: 2

Explanation: The subarray `[4,3]` has the minimal length under the problem constraint.

Example 2:

Input: `target = 4, nums = [1,4,4]`

Output: 1

Example 3:

Input: `target = 11, nums = [1,1,1,1,1,1,1,1]`

Output: 0

Constraints:

- $1 \leq \text{target} \leq 10^9$
- $1 \leq \text{nums.length} \leq 10^5$

- $1 \leq \text{nums}[i] \leq 10^4$

3. Longest Substring Without Repeating Characters

Solved

Medium

Topics

Companies

Hint

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

Example 1:

Input: $s = \text{"abcabcbb"}$

Output: 3

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

Example 2:

Input: $s = \text{"bbbbbb"}$

Output: 1

Explanation: The answer is "b", with the length of 1.

Example 3:

Input: $s = \text{"pwwkew"}$

Output: 3

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

Notice that the answer must be a substring, "pwke" is a subsequence and not a substring.

Constraints:

- $0 \leq s.\text{length} \leq 5 * 10^4$
- s consists of English letters, digits, symbols and spaces.

36. Valid Sudoku

Solved

Medium

Topics

Companies

Determine if a 9 x 9 Sudoku board is valid. Only the filled cells need to be validated **according to the following rules**:

1. Each row must contain the digits 1-9 without repetition.
2. Each column must contain the digits 1-9 without repetition.
3. Each of the nine 3 x 3 sub-boxes of the grid must contain the digits 1-9 without repetition.

Note:

- A Sudoku board (partially filled) could be valid but is not necessarily solvable.
- Only the filled cells need to be validated according to the mentioned rules.

Example 1:

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Input: board =

```
[["5","3",".",".","7",".",".",".","."],  
["6",".",".","1","9","5",".",".","."],  
[".","9","8",".",".",".","6","."],  
["8",".",".","6",".",".","3"],  
["4",".","8",".","3",".","1"],  
["7",".","2",".","6"],  
[".","6",".","2","8","."],  
[".","4","1","9",".","5"],  
[".","8",".","7","9"]]
```

Output: true

Example 2:

Input: board =

```
[["8","3",".",".","7",".",".","."],
["6",".",".","1","9","5",".","."],
[".","9","8",".",".",".","6","."],
["8",".",".","6",".",".","3"],
["4",".","8",".","3",".","1"],
["7",".","2",".",".","6"],
[".","6",".","2","8","."],
[".","4","1","9",".","5"],
[".","8",".","7","9"]]
```

Output: false

Explanation: Same as Example 1, except with the 5 in the top left corner being modified to 8. Since there are two 8's in the top left 3x3 sub-box, it is invalid.

Constraints:

- board.length == 9
 - board[i].length == 9
 - board[i][j] is a digit 1-9 or '.'.
-

54. Spiral Matrix

Solved

Medium

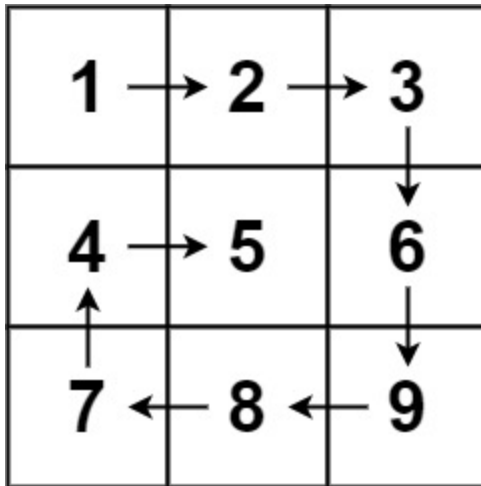
Topics

Companies

Hint

Given an **m x n** matrix, return *all elements of the matrix in spiral order.*

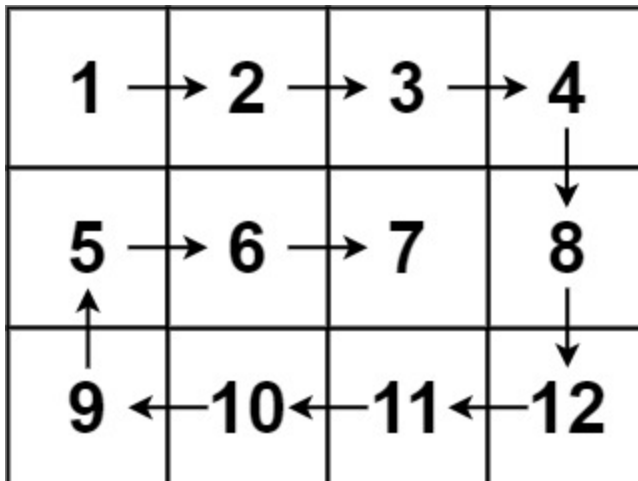
Example 1:



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

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

Example 2:



Input: matrix = [[1,2,3,4],[5,6,7,8],[9,10,11,12]]

Output: [1,2,3,4,8,12,11,10,9,5,6,7]

Constraints:

- m == matrix.length
- n == matrix[i].length
- 1 <= m, n <= 10
- -100 <= matrix[i][j] <= 100

48. Rotate Image

26

Solved

Medium

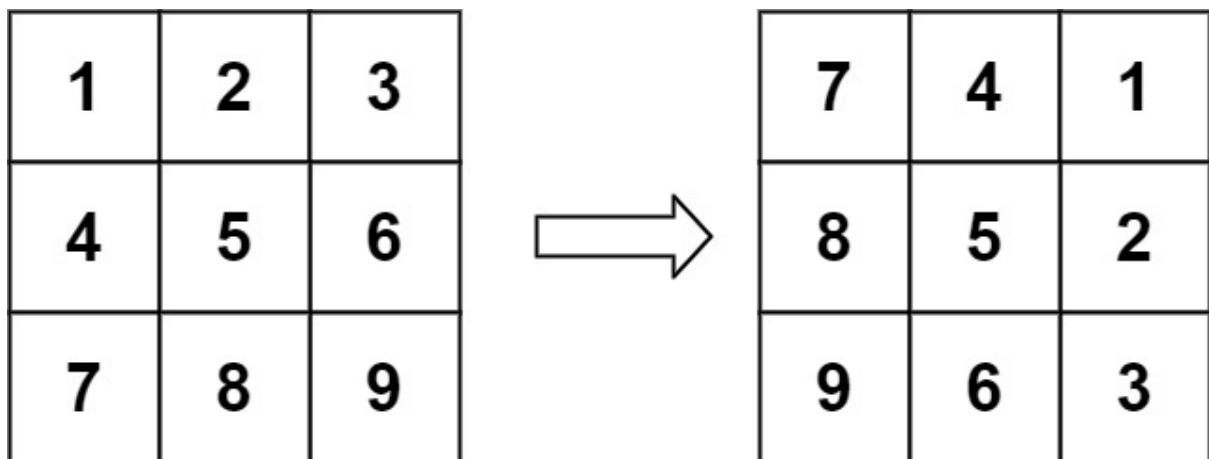
Topics

Companies

You are given an $n \times n$ 2D matrix representing an image, rotate the image by **90** degrees (clockwise).

You have to rotate the image **in-place**, which means you have to modify the input 2D matrix directly. **DO NOT** allocate another 2D matrix and do the rotation.

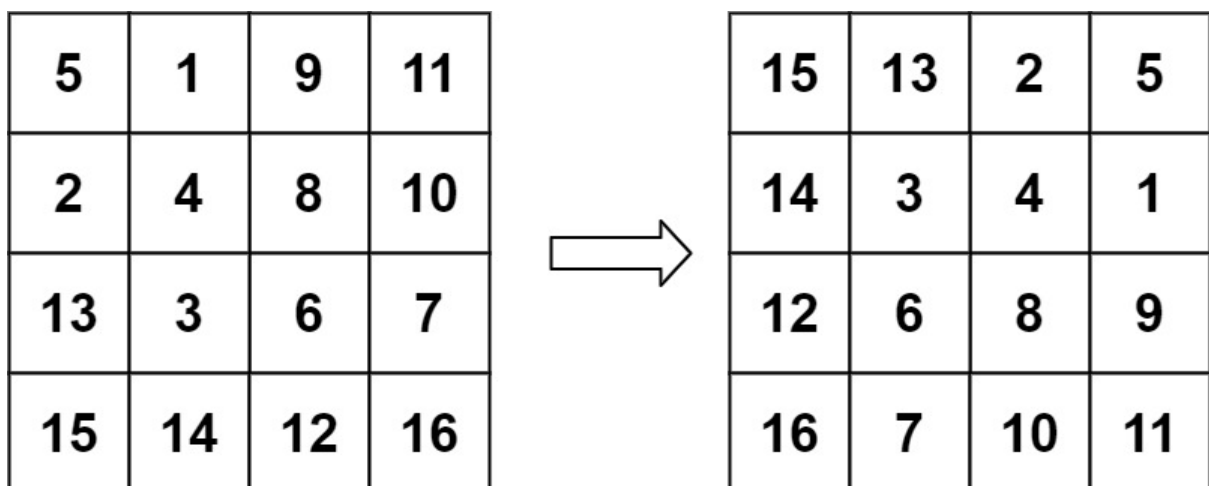
Example 1:



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

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

Example 2:



Input: matrix = [[5,1,9,11],[2,4,8,10],[13,3,6,7],[15,14,12,16]]

Output: [[15,13,2,5],[14,3,4,1],[12,6,8,9],[16,7,10,11]]

Constraints:

- `n == matrix.length == matrix[i].length`
- `1 <= n <= 20`
- `-1000 <= matrix[i][j] <= 1000`

73. Set Matrix Zeroes

21.

Solved

Medium

Topics

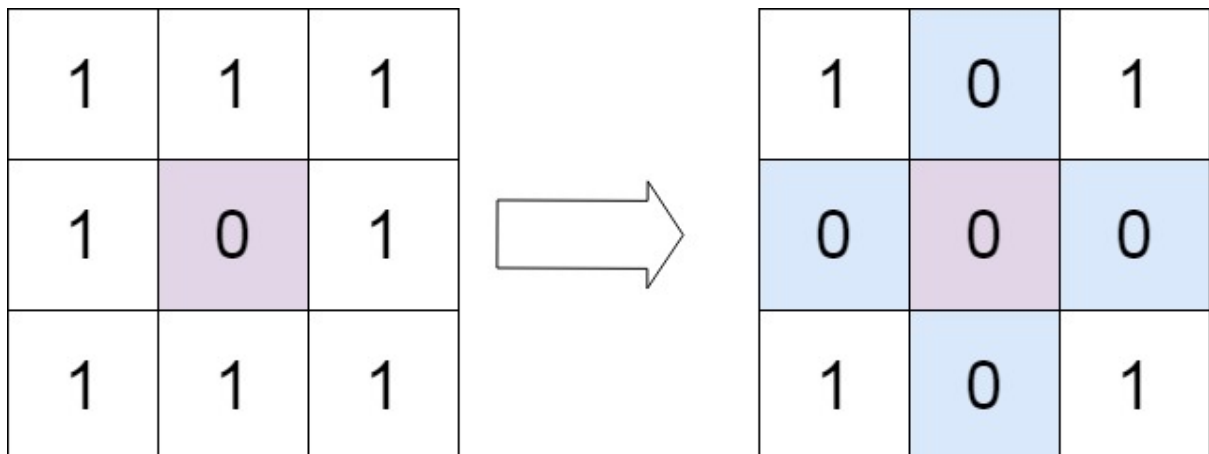
Companies

Hint

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

You must do it [in place](#).

Example 1:

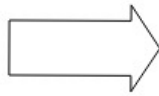


Input: `matrix = [[1,1,1],[1,0,1],[1,1,1]]`

Output: `[[1,0,1],[0,0,0],[1,0,1]]`

Example 2:

0	1	2	0
3	4	5	2
1	3	1	5



0	0	0	0
0	4	5	0
0	3	1	0

Input: matrix = [[0,1,2,0],[3,4,5,2],[1,3,1,5]]

Output: [[0,0,0,0],[0,4,5,0],[0,3,1,0]]

Constraints:

- `m == matrix.length`
- `n == matrix[0].length`
- $1 \leq m, n \leq 200$
- $-2^{31} \leq \text{matrix}[i][j] \leq 2^{31} - 1$

289. Game of Life

Solved

Medium

Topics

Companies

According to [Wikipedia's article](#): "The **Game of Life**, also known simply as **Life**, is a cellular automaton devised by the British mathematician John Horton Conway in 1970."

The board is made up of an $m \times n$ grid of cells, where each cell has an initial state: **live** (represented by a 1) or **dead** (represented by a 0). Each cell interacts with its [eight neighbors](#) (horizontal, vertical, diagonal) using the following four rules (taken from the above Wikipedia article):

1. Any live cell with fewer than two live neighbors dies as if caused by under-population.
2. Any live cell with two or three live neighbors lives on to the next generation.
3. Any live cell with more than three live neighbors dies, as if by over-population.
4. Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

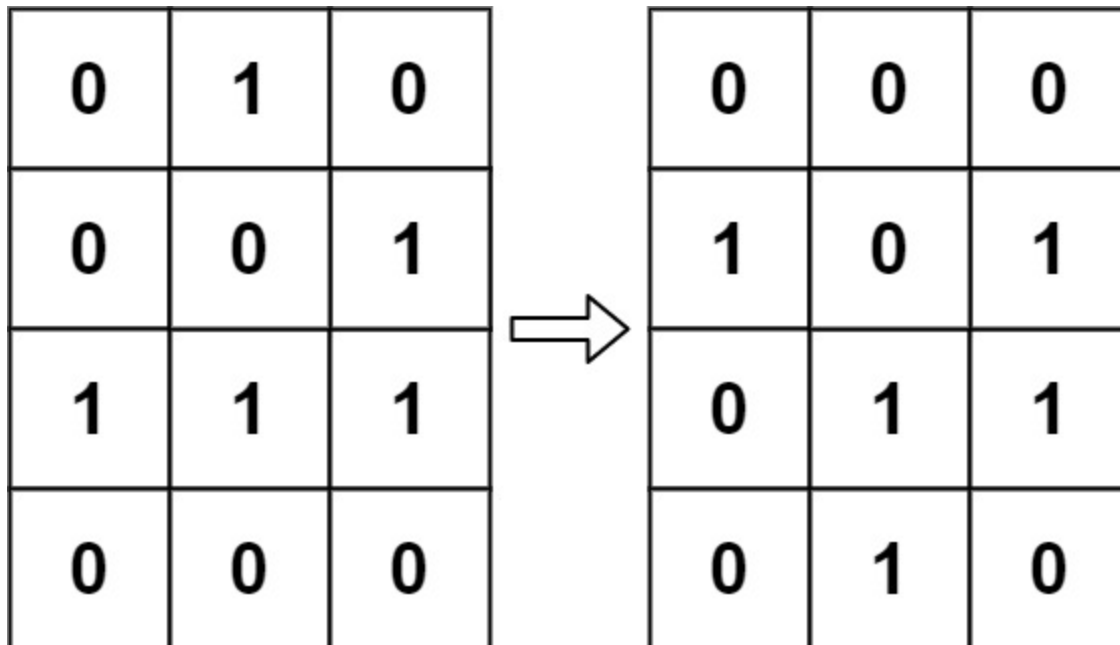
The next state of the board is determined by applying the above rules simultaneously to every cell in the current state of the $m \times n$ grid board. In this process, births and deaths occur **simultaneously**.

22

Given the **current state** of the board, **update** the board to reflect its next state.

Note that you do not need to return anything.

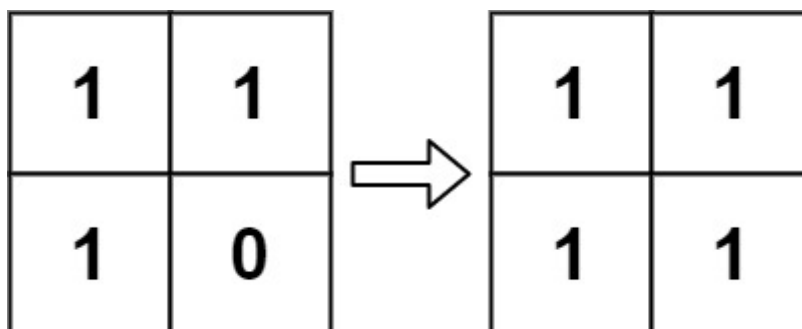
Example 1:



Input: board = [[0,1,0],[0,0,1],[1,1,1],[0,0,0]]

Output: [[0,0,0],[1,0,1],[0,1,1],[0,1,0]]

Example 2:



Input: board = [[1,1],[1,0]]

Output: [[1,1],[1,1]]

Constraints:

- $m == \text{board.length}$
- $n == \text{board}[i].\text{length}$
- $1 \leq m, n \leq 25$

- `board[i][j]` is 0 or 1.

Testcase

Test Result

Test Result

Code

49. Group Anagrams

Solved

Medium

Topics

Companies

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

Example 1:

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

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

Explanation:

- There is no string in `strs` that can be rearranged to form "bat".
- The strings "nat" and "tan" are anagrams as they can be rearranged to form each other.
- The strings "ate", "eat", and "tea" are anagrams as they can be rearranged to form each other.

Example 2:

Input: `strs = [""]`

Output: `[[""]]`

Example 3:

Input: `strs = ["a"]`

Output: `[["a"]]`

Constraints:

- $1 \leq \text{strs.length} \leq 10^4$

- $0 \leq \text{strs}[i].\text{length} \leq 100$
- $\text{strs}[i]$ consists of lowercase English letters.

56. Merge Intervals

Solved

Medium

Topics

Companies

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 1:

Input: $\text{intervals} = [[1,3],[2,6],[8,10],[15,18]]$

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

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

Example 2:

Input: $\text{intervals} = [[1,4],[4,5]]$

Output: $[[1,5]]$

Explanation: Intervals $[1,4]$ and $[4,5]$ are considered overlapping.

Constraints:

- $1 \leq \text{intervals.length} \leq 10^4$
- $\text{intervals}[i].\text{length} == 2$
- $0 \leq \text{start}_i \leq \text{end}_i \leq 10^4$

57. Insert Interval

Solved

Medium

Topics

Companies

Hint

You are given an **array of non-overlapping intervals** intervals where intervals[i] = [start_i, end_i] represent the start and the **end of the ith interval** and intervals is sorted in ascending order **by start_i**. 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 start_i and intervals still does not have any **overlapping intervals** (merge overlapping intervals if necessary).

Return intervals *after the insertion*.

Note that you don't need to modify **intervals in-place**. You can make a new array and return it.

Example 1:

Input: intervals = [[1,3],[6,9]], newInterval = [2,5]

Output: [[1,5],[6,9]]

Example 2:

Input: intervals = [[1,2],[3,5],[6,7],[8,10],[12,16]], newInterval = [4,8]

Output: [[1,2],[3,10],[12,16]]

Explanation: Because the new interval [4,8] overlaps with [3,5],[6,7],[8,10].

Constraints:

- $0 \leq \text{intervals.length} \leq 10^4$
- $\text{intervals}[i].\text{length} == 2$
- $0 \leq \text{start}_i \leq \text{end}_i \leq 10^5$
- intervals is sorted by start_i in **ascending** order.
- $\text{newInterval.length} == 2$
- $0 \leq \text{start} \leq \text{end} \leq 10^5$

452. Minimum Number of Arrows to Burst Balloons

Solved

Medium

Topics

Companies

There are some spherical balloons taped onto a flat wall that represents the XY-plane. The balloons are represented as a 2D integer array points where points[i] = [x_{start}, x_{end}] denotes a balloon

26

whose **horizontal diameter** stretches between x_{start} and x_{end} . You do not know the exact y-coordinates of the balloons.

Arrows can be shot up **directly vertically** (in the positive y-direction) from different points along the x-axis. A balloon with x_{start} and x_{end} is **burst** by an arrow shot at x if $x_{start} \leq x \leq x_{end}$. There is **no limit** to the number of arrows that can be shot. A shot arrow keeps traveling up infinitely, bursting any balloons in its path.

Given the array points, return *the **minimum** number of arrows that must be shot to burst all balloons.*

Example 1:

Input: points = [[10,16],[2,8],[1,6],[7,12]]

Output: 2

Explanation: The balloons can be burst by 2 arrows:

- Shoot an arrow at $x = 6$, bursting the balloons [2,8] and [1,6].
- Shoot an arrow at $x = 11$, bursting the balloons [10,16] and [7,12].

Example 2:

Input: points = [[1,2],[3,4],[5,6],[7,8]]

Output: 4

Explanation: One arrow needs to be shot for each balloon for a total of 4 arrows.

Example 3:

Input: points = [[1,2],[2,3],[3,4],[4,5]]

Output: 2

Explanation: The balloons can be burst by 2 arrows:

- Shoot an arrow at $x = 2$, bursting the balloons [1,2] and [2,3].
- Shoot an arrow at $x = 4$, bursting the balloons [3,4] and [4,5].

Constraints:

- $1 \leq \text{points.length} \leq 10^5$
- $\text{points}[i].\text{length} == 2$
- $-2^{31} \leq x_{start} < x_{end} \leq 2^{31} - 1$

Testcase

Test Result

Test Result

Code

71. Simplify Path

Solved

Medium

Topics

Companies

You are given an *absolute* path for a Unix-style file system, which always begins with a slash '/'. Your task is to transform this absolute path into its **simplified canonical path**.

The *rules* of a Unix-style file system are as follows:

- A single period '.' represents the current directory.
- A double period '..' represents the previous/parent directory.
- Multiple consecutive slashes such as '/' and '//' are treated as a single slash '/'.
- Any sequence of periods that does **not match** the rules above should be treated as a **valid directory or file name**. For example, '...' and '....' are valid directory or file names.

The simplified canonical path should follow these *rules*:

- The path must start with a single slash '/'.
- Directories within the path must be separated by exactly one slash '/'.
- The path must not end with a slash '/', unless it is the root directory.
- The path must not have any single or double periods ('.' and '..') used to denote current or parent directories.

Return the **simplified canonical path**.

Example 1:

Input: path = "/home/"

Output: "/home"

Explanation:

The trailing slash should be removed.

Example 2:

Input: path = "/home//foo/"

Output: "/home/foo"

Explanation:

Multiple consecutive slashes are replaced by a single one.

Example 3:

Input: path = "/home/user/Documents/../Pictures"

Output: "/home/user/Pictures"

Explanation:

A double period ".." refers to the directory up a level (the parent directory).

Example 4:

Input: path = "/../"

Output: "/"

Explanation:

Going one level up from the root directory is not possible.

Example 5:

Input: path = "/.../a/../../b/c/../../d/.."

Output: "/.../b/d"

Explanation:

"..." is a valid name for a directory in this problem.

Constraints:

- $1 \leq \text{path.length} \leq 3000$
 - path consists of English letters, digits, period '.', slash '/' or '_'.
 - path is a valid absolute Unix path.
-

155. Min Stack

Solved

Medium

Topics

Companies

Hint

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

Implement the MinStack class:

- MinStack() initializes the stack object.
- void push(int val) pushes the element val onto the stack.

28

- void pop() removes the element on the top of the stack.
- int top() gets the top element of the stack.
- int getMin() retrieves the minimum element in the stack.

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

Example 1:

Input

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

Output

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

Explanation

```
MinStack minStack = new MinStack();
minStack.push(-2);
minStack.push(0);
minStack.push(-3);
minStack.getMin(); // return -3
minStack.pop();
minStack.top();    // return 0
minStack.getMin(); // return -2
```

Constraints:

- $-2^{31} \leq \text{val} \leq 2^{31} - 1$
- Methods pop, top and getMin operations will always be called on **non-empty** stacks.
- At most $3 \cdot 10^4$ calls will be made to push, pop, top, and getMin.

150. Evaluate Reverse Polish Notation

Solved

Medium

Topics

Companies

You are given an array of strings tokens that represents an arithmetic expression in a [Reverse Polish Notation](#).

Evaluate the expression. Return *an integer that represents the value of the expression*.

Note that:

- The valid operators are '+', '-', '*', and '/'.
- Each operand may be an integer or another expression.
- The division between two integers always **truncates toward zero**.
- There will not be any division by zero.
- The input represents a valid arithmetic expression in a reverse polish notation.
- The answer and all the intermediate calculations can be represented in a **32-bit** integer.

Example 1:

Input: tokens = ["2","1","+","3","*"]

Output: 9

Explanation: $((2 + 1) * 3) = 9$

Example 2:

Input: tokens = ["4","13","5","/","+"]

Output: 6

Explanation: $(4 + (13 / 5)) = 6$

Example 3:

Input: tokens = ["10","6","9","3","+","-11","*","/", "*","17","+","5","+"]

Output: 22

Explanation: $((10 * (6 / ((9 + 3) * -11))) + 17) + 5$

$= ((10 * (6 / (12 * -11))) + 17) + 5$

$= ((10 * (6 / -132)) + 17) + 5$

$= ((10 * 0) + 17) + 5$

$= (0 + 17) + 5$

$= 17 + 5$

$= 22$

Constraints:

- $1 \leq \text{tokens.length} \leq 10^4$
 - `tokens[i]` is either an operator: "+", "-", "*", or "/", or an integer in the range [-200, 200].
-