Description | Editorial | Solutions | Submissions

1572. Matrix Diagonal Sum

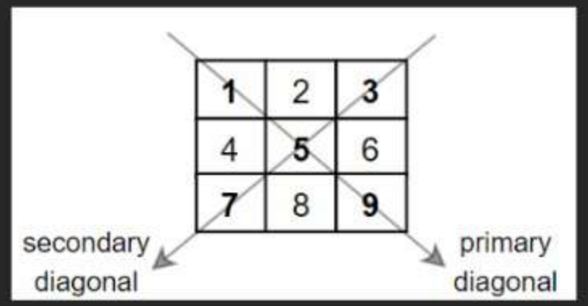
Solved

Easy | Topics | Companies | Hint

Given a square matrix `mat`, return the sum of the matrix diagonals.

Only include the sum of all the elements on the primary diagonal and all the elements on the secondary diagonal that are not part of the primary diagonal.

Example 1:




```
Input: mat = [[1,2,3],
              [4,5,6],
              [7,8,9]]
Output: 25
Explanation: Diagonals sum: 1 + 5 + 9 + 3 + 7 = 25
Notice that element mat[1][1] = 5 is counted only once.
```

Example 2:

```
Input: mat = [[1,1,1,1],
```

3.8K | 66 | 25 Online

Code

Java | Auto

```java
class Solution {
    public int diagonalSum(int[][] mat) {
        int n = mat.length;
        int sum = 0;

        for (int i = 0; i < n; i++) {
            sum += mat[i][i];
            sum += mat[i][n - 1 - i];
        }
        if (n % 2 == 1) {
            sum -= mat[n / 2][n / 2];
        }

        return sum;
    }
}
```

Saved | Ln 1, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

mat =
[[1,2,3],[4,5,6],[7,8,9]]

Output

25

Description | Editorial | Solutions | Submissions

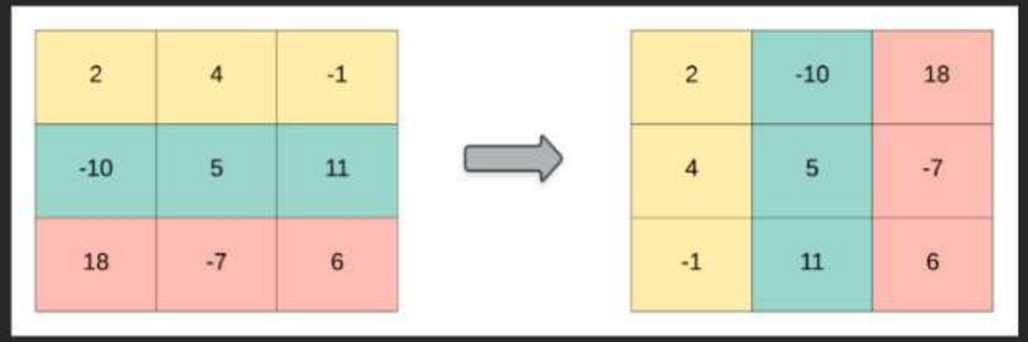
867. Transpose Matrix

Solved

Easy | Topics | Companies | Hint

Given a 2D integer array `matrix`, return *the **transpose** of* `matrix`.

The **transpose** of a matrix is the matrix flipped over its main diagonal, switching the matrix's row and column indices.



Example 1:

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

Example 2:

```
Input: matrix = [[1,2,3],[4,5,6]]
Output: [[1,4],[2,5],[3,6]]
```

4.1K | 128 | 26 Online

Run Ctrl

Java | Auto

```java
class Solution {
    public int[][] transpose(int[][] matrix) {
        int m = matrix.length;
        int n = matrix[0].length;

        int[][] t = new int[n][m];

        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                t[j][i] = matrix[i][j];
            }
        }

        return t;
    }
}
```

Saved | Ln 1, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

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

Output

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

Description | Editorial | Solutions | Submissions

153. Find Minimum in Rotated Sorted Array

Medium | Topics | Companies | Hint

Suppose an array of length `n` sorted in ascending order is **rotated** between `1` and `n` times. For example, the array `nums = [0,1,2,4,5,6,7]` might become:

- `[4,5,6,7,0,1,2]` if it was rotated `4` times.
- `[0,1,2,4,5,6,7]` if it was rotated `7` times.

Notice that **rotating** an array `[a[0], a[1], a[2], ..., a[n-1]]` 1 time results in the array `[a[n-1], a[0], a[1], a[2], ..., a[n-2]]`.

Given the sorted rotated array `nums` of **unique** elements, return *the minimum element of this array*.

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

Example 1:

```
Input: nums = [3,4,5,1,2]
Output: 1
Explanation: The original array was [1,2,3,4,5]
rotated 3 times.
```

Example 2:

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


Code

Java | Auto

```java
class Solution {
    public int findMin(int[] nums) {
        int l = 0, r = nums.length - 1;

        while (l < r) {
            int m = (l + r) / 2;

            if (nums[m] > nums[r]) {
                l = m + 1;      // min in right half
            } else {
                r = m;          // min in left half (including m)
            }
        }

        return nums[l];
    }
}
```

Saved | Ln 18, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

nums =
[3,4,5,1,2]

Output

1

Description | Editorial | Solutions | Submissions

28. Find the Index of the First Occurrence in a String

Easy | Topics | Companies

Given two strings `needle` and `haystack`, return the index of the first occurrence of `needle` in `haystack`, or `-1` if `needle` is not part of `haystack`.

Example 1:

```
Input: haystack = "sadbutsad", needle = "sad"
Output: 0
Explanation: "sad" occurs at index 0 and 6.
The first occurrence is at index 0, so we return 0.
```

Example 2:

```
Input: haystack = "leetcode", needle = "leeto"
Output: -1
Explanation: "leeto" did not occur in "leetcode", so we return -1.
```

Constraints:

- `1 <= haystack.length, needle.length <= 10^4`
- `haystack` and `needle` consist of only lowercase English characters.

Code

Java | Auto

```java
class Solution {
    public int strStr(String haystack, String needle) {
        int n = haystack.length();
        int m = needle.length();

        for (int i = 0; i <= n - m; i++) {
            int j = 0;

            while (j < m && haystack.charAt(i + j) == needle.charAt(j)) {
                j++;
            }

            if (j == m) return i;
        }

        return -1;
    }
}

```

Saved | Ln 19, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

haystack =
"sadbutsad"

needle =

Description | Accepted | Editorial | Solutions | Submissic

704. Binary Search

Solved

Easy | Topics | Companies

Given an array of integers `nums` which is sorted in ascending order, and an integer `target`, write a function to search `target` in `nums`. If `target` exists, then return its index. Otherwise, return `-1`.

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

Example 1:

```
Input: nums = [-1,0,3,5,9,12], target = 9
Output: 4
Explanation: 9 exists in nums and its index is 4
```

Example 2:

```
Input: nums = [-1,0,3,5,9,12], target = 2
Output: -1
Explanation: 2 does not exist in nums so return -1
```

Constraints:

- `1 <= nums.length <= 10^4`
- `-10^4 < nums[i], target < 10^4`
- All the integers in `nums` are **unique**.

13.3K | 206 | 269 Online

Code

Java | Auto

```java
class Solution {
    public int search(int[] nums, int target) {
        int l = 0, r = nums.length - 1;

        while (l <= r) {
            int m = (l + r) / 2;

            if (nums[m] == target) return m;
            else if (nums[m] < target) l = m + 1;
            else r = m - 1;
        }

        return -1;
    }
}
```

Saved | Ln 16, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

nums =
[-1,0,3,5,9,12]

target =
9

Description | Accepted | Editorial | Solutions | Submissions

278. First Bad Version

Solved

Easy | Topics | Companies

You are a product manager and currently leading a team to develop a new product. Unfortunately, the latest version of your product fails the quality check. Since each version is developed based on the previous version, all the versions after a bad version are also bad.

Suppose you have `n` versions `[1, 2, ..., n]` and you want to find out the first bad one, which causes all the following ones to be bad.

You are given an API `bool isBadVersion(version)` which returns whether `version` is bad. Implement a function to find the first bad version. You should minimize the number of calls to the API.

Example 1:

```
Input: n = 5, bad = 4
Output: 4
Explanation:
call isBadVersion(3) -> false
call isBadVersion(5) -> true
call isBadVersion(4) -> true
Then 4 is the first bad version.
```

Example 2:

```
Input: n = 1, bad = 1
Output: 1
```

9K | 280 | 74 Online

Code

Java | Auto

```java
public class Solution extends VersionControl {
    public int firstBadVersion(int n) {
        int l = 1, r = n;

        while (l < r) {
            int m = l + (r - l) / 2;

            if (isBadVersion(m)) {
                r = m;
            } else {
                l = m + 1;
            }
        }
        return l;
    }
}
```

Saved | Ln 13, Col 10

Testcase | Test Result

Accepted Runtime: 1 ms

Case 1 | Case 2

Input

n =
5

bad =
4

Description | Accepted | Editorial | Solutions

35. Search Insert Position

Solved

Easy | Topics | Companies

Given a sorted array of distinct integers and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order.

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

Example 1:

```
Input: nums = [1,3,5,6], target = 5
Output: 2
```

Example 2:

```
Input: nums = [1,3,5,6], target = 2
Output: 1
```

Example 3:

```
Input: nums = [1,3,5,6], target = 7
Output: 4
```

Constraints:

- `1 <= nums.length <= 10^4`
- `-10^4 <= nums[i] <= 10^4`

18.5K | 414 | 268 Online

Code

Java | Auto

```java
class Solution {
    public int searchInsert(int[] nums, int target) {
        int l = 0;
        int r = nums.length - 1;
        while (l <= r) {
            int m = (l + r) / 2;
            if (nums[m] == target) {
                return m;
            }
            else if (nums[m] < target) {
                l = m + 1;
            }
            else {
                r = m - 1;
            }
        }
        return l;
    }
}
```

Saved | Ln 16, Col 10

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

nums =

[1,3,5,6]

Description | Accepted | Editorial | Solutions | Submiss

34. Find First and Last Position of Element in Sorted Array

Solved

Medium | Topics | Companies

Given an array of integers `nums` sorted in non-decreasing order, find the starting and ending position of a given `target` value.

If `target` is not found in the array, return `[-1, -1]`.

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

Example 1:

```
Input: nums = [5,7,7,8,8,10], target = 8
Output: [3,4]
```

Example 2:

```
Input: nums = [5,7,7,8,8,10], target = 6
Output: [-1,-1]
```

Example 3:

```
Input: nums = [], target = 0
Output: [-1,-1]
```

Constraints:

Code

Java | Auto

```java
class Solution {
    public int[] searchRange(int[] nums, int target) {
        int[] res = {-1, -1};
        res[0] = firstPos(nums, target);
        res[1] = lastPos(nums, target);
        return res;
    }
    private int firstPos(int[] a, int t) {
        int l = 0, r = a.length - 1;
        int ans = -1;
        while (l <= r) {
            int m = (l + r) / 2;
            if (a[m] == t) {
                ans = m;
                r = m - 1;
            } else if (a[m] < t) {
```

Saved | Ln 34, Col 29

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

nums =
[5,7,7,8,8,10]

target =
8

Description | Editorial | Solutions | Submissions

1672. Richest Customer Wealth

Solved

Easy | Topics | Companies | Hint

You are given an `m x n` integer grid `accounts` where `accounts[i][j]` is the amount of money the i^{th} customer has in the j^{th} bank. Return *the* ***wealth*** *that the richest customer has.*

A customer's **wealth** is the amount of money they have in all their bank accounts. The richest customer is the customer that has the maximum **wealth**.

Example 1:

> **Input:** accounts = [[1,2,3],[3,2,1]]
> **Output:** 6
> **Explanation:**
> 1st customer has wealth = 1 + 2 + 3 = 6
> 2nd customer has wealth = 3 + 2 + 1 = 6
> Both customers are considered the richest with a wealth of 6 each, so return 6.

Example 2:

> **Input:** accounts = [[1,5],[7,3],[3,5]]
> **Output:** 10
> **Explanation:**
> 1st customer has wealth = 6
> 2nd customer has wealth = 10
> 3rd customer has wealth = 8
> The 2nd customer is the richest with a wealth of 10.

Example 3:

Code

Java | Auto

```java
class Solution {
    public int maximumWealth(int[][] accounts) {
        int max = 0;
        for (int i = 0; i < accounts.length; i++) {
            int sum = 0;
            for (int j = 0; j < accounts[i].length; j++) {
                sum += accounts[i][j];
            }
            if (sum > max) {
                max = sum;
            }
        }

        return max;
    }
}
```

Saved | Ln 1, Col 1

Testcase | Test Result

You must run your code first