

Description

Editorial

Solutions (6.1K)

Submissions

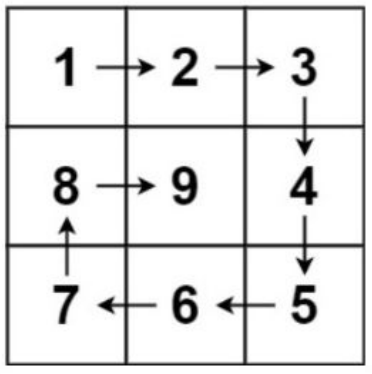
59. Spiral Matrix II

Medium✔️👍 5.9K💬 239★🔄

🏢 Companies

Given a positive integer n , generate an $n \times n$ matrix filled with elements from 1 to n^2 in spiral order.

Example 1:



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

Example 2:

i Java

• Auto

🔖 {} ↺ ⚙️ ↗️

```
1 class Solution {
2     public int[][] generateMatrix(int n) {
3         int[][] ans = new int[n][n];
4         int top = 0, left = 0;
5         int bottom = n-1, right = n-1;
6         int k = 1;
7         while(top<=bottom && left<=right){
8             for(int i=left;i<=right;i++){
9                 ans[top][i] = k++;
10            }
11            top++;
12            for(int i=top;i<=bottom;i++){
13                ans[i][right] = k++;
14            }
15            right--;
16            if(top<=bottom){
17                for(int i=right;i>=left;i--){
18                    ans[bottom][i] = k++;
19                }
20                bottom--;
21            }
22            if(left<=right){
23                for(int i=bottom;i>=top;i--){
24                    ans[i][left] = k++;
25                }
26                left++;
27            }
28        }
29        return ans;
30    }
31 }
```

1672. Richest Customer Wealth

Hint

Easy
3.7K
338

Companies

You are given an $m \times n$ 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:

<https://leetcode.com/problems/richest-customer-wealth/description>

i Java • Auto

```

1 class Solution {
2     public int maximumWealth(int[][] accounts) {
3         int sum = 0;
4         for(int i=0; i<accounts.length; i++){
5             int temp_wealth = 0;
6             for(int j=0; j<accounts[i].length; j++){
7                 temp_wealth +=accounts[i][j];
8             }
9             sum = Math.max(sum, temp_wealth);
10        }
11        return sum;
12    }
13 }
```

Console

Run

Submit



766. Toeplitz Matrix

Easy 3.3K 155

Companies

Given an $m \times n$ matrix, return *true* if the matrix is Toeplitz. Otherwise, return *false*.

A matrix is **Toeplitz** if every diagonal from top-left to bottom-right has the same elements.

Example 1:

1	2	3	4
5	1	2	3
9	5	1	2

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

Output: true

<https://leetcode.com/problems/toeplitz-matrix/description>

Hint

i Java Auto

```

1 class Solution {
2     public boolean isToeplitzMatrix(int[][] matrix) {
3         for(int i=1; i<matrix.length; i++){
4             for(int j=1; j<matrix[0].length; j++){
5                 if(matrix[i][j] != matrix[i-1][j-1])
6                     return false;
7             }
8         }
9         return true;
10    }
11 }
```

Console

Run

Run

Submit

378. Kth Smallest Element in a Sorted Matrix

Medium 9.3K 320

Companies

Given an $n \times n$ matrix where each of the rows and columns is sorted in ascending order, return the k^{th} smallest element in the matrix.

Note that it is the k^{th} smallest element in the sorted order, not the k^{th} distinct element.

You must find a solution with a memory complexity better than $O(n^2)$.

Example 1:

Input: matrix = [[1,5,9],[10,11,13],[12,13,15]], k = 8

Output: 13

Explanation: The elements in the matrix are [1,5,9,10,11,12,13,13,15], and the 8th smallest number is 13

Example 2:

Input: matrix = [[-5]], k = 1

Output: -5

```
1 class Solution {
2     public int kthSmallest(int[][] matrix, int k) {
3         Map<Integer, Integer> map = new TreeMap<>();
4
5         for(int i = 0; i < matrix.length; i++) {
6             for(int j = 0; j < matrix[0].length; j++) {
7                 map.put(matrix[i][j], map.getDefault(matrix[i][j], 0) + 1);
8             }
9         }
10
11         for(int n : map.keySet()) {
12             k -= map.get(n);
13
14             if(k <= 0) {
15                 return n;
16             }
17         }
18
19         return -1;
20     }
21 }
```

Description

Editorial

Solutions (5K)

Submissions

1572. Matrix Diagonal Sum

Hint

Easy

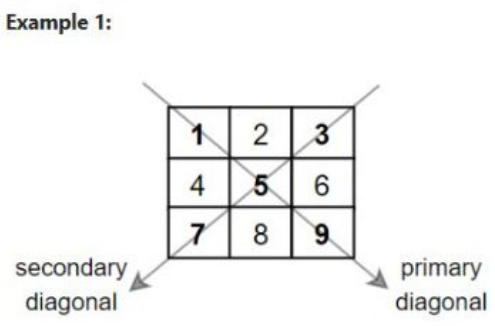
3.1K

41

Companies

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.



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.

i Java

Auto

```
1 class Solution {
2     public int diagonalSum(int[][] mat) {
3         int verticalSum=0;
4         for(int i=0; i<mat.length; i++){
5             verticalSum+=mat[i][i];
6         }
7         int count=0;
8         int horizontalSum=0;
9         for(int j=mat[0].length-1; j>=0; j--){
10             if(mat[0].length/2==count && mat[0].length%2!=0){
11                 count++;
12                 continue;
13             }
14             horizontalSum+=mat[count][j];
15             count++;
16         }
17         return (verticalSum+horizontalSum);
18     }
19 }
```

Testcase

Result

Accepted

Runtime: 0 ms

Case 1

Case 2

Case 3

Console

Run

Submit

1351. Count Negative Numbers in a Sorted Matrix

Hint

Easy

 4.6K
 119

Companies

Given a $m \times n$ matrix `grid` which is sorted in non-increasing order both row-wise and column-wise, return the number of **negative** numbers in `grid`.

Example 1:

Input: `grid = [[4,3,2,-1],[3,2,1,-1],[1,1,-1,-2],[-1,-1,-2,-3]]`
Output: 8
Explanation: There are 8 negatives number in the matrix.

Example 2:

Input: `grid = [[3,2],[1,0]]`
Output: 0

Constraints:

- `m == grid.length`

<https://leetcode.com/problems/count-negative-numbers-in-a-sorted-matrix/description>

i Java • Auto

```

1 class Solution {
2     public int countNegatives(int[][] grid) {
3         int ans=0;
4         int n = grid.length;
5         int m = grid[0].length;
6         for(int i = 0; i < n; i++){
7             for(int j = 0; j < m; j++){
8                 if(grid[i][j]<0) ans++;
9             }
10        }
11        return ans;
12    }
13 }
```

Console

Run

Submit



867. Transpose Matrix

Hint

Easy 2.9K 425

Companies

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.

2	4	-1
-10	5	11
18	-7	6

→

2	-10	18
4	5	-7
-1	11	6

Example 1:

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

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

<https://leetcode.com/problems/transpose-matrix/description>

i Java Auto

{} ↺ ⚙️ ↻

```

1 class Solution {
2     public int[][] transpose(int[][] matrix) {
3         int[][] answer = new int[matrix[0].length][matrix.length];
4         for(int i=0; i < matrix.length; i++){
5             for (int j = 0; j < matrix[0].length; j++){
6                 answer[j][i] = matrix[i][j];
7             }
8         }
9         return answer;
10    }
11 }

```

Console ^

⚙️

Run

Submit



73. Set Matrix Zeroes

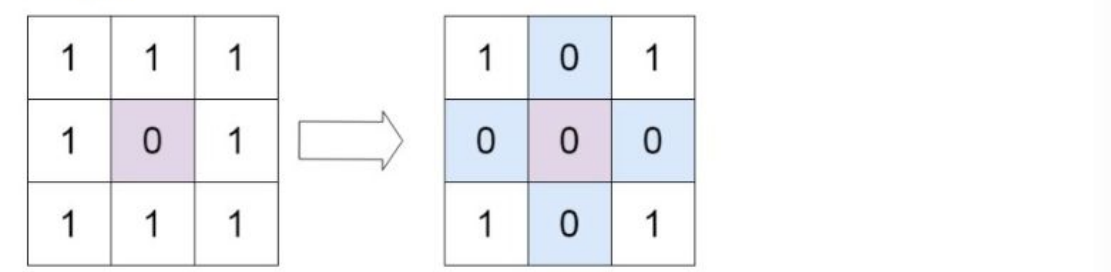
Hint

Medium 12.5K 639

Companies

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:

`[1,0,1,0]`
`[0,1,0,1]`
`[1,1,1,1]`
`[0,0,0,0]`

```
1 class Solution {
2     public void setZeroes(int[][] matrix) {
3         Set<Integer> row = new HashSet<>();
4         Set<Integer> col = new HashSet<>();
5         for(int i = 0; i < matrix.length; i++){
6             for(int j = 0; j < matrix[0].length; j++){
7                 if(matrix[i][j] == 0){
8                     row.add(i);
9                     col.add(j);
10                }
11            }
12        }
13        for(int r : row){
14            for(int i = 0; i < matrix[0].length; i++){
15                matrix[r][i] = 0;
16            }
17        }
18        for(int c : col){
19            for(int i = 0; i < matrix.length; i++){
20                matrix[i][c] = 0;
21            }
22        }
23    }
24 }
```