

← C https://www.geeksforgeeks.org/problems/spirally-traversing-a-matrix-1587115621/1 ⏪ ⏴ ⏵ ⏷ Chat

Get 90% Refund Courses Tutorials Practice Jobs

Search... Problem Editorial Submissions Comments Java (21) Start Timer

You are given a rectangular matrix `mat[][]` of size `n x m`, and your task is to return an array while traversing the matrix in **spiral** form.

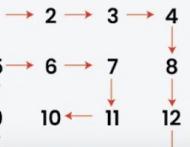
Examples:

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

Explanation:

Example of matrix in spiral form

Matrix:



```
1 * import java.util.*;  
2 *  
3 * class Solution {  
4 *     public ArrayList<Integer> spirallyTraverse(int mat[][]) {  
5 *         ArrayList<Integer> result = new ArrayList<>();  
6 *         if (mat == null || mat.length == 0) return result;  
7 *  
8 *         int n = mat.length;  
9 *         int m = mat[0].length;  
10 *  
11 *         int top = 0, bottom = n - 1;  
12 *         int left = 0, right = m - 1;  
13 *  
14 *         while (top <= bottom && left <= right) {  
15 *             // 1. Traverse from left to right across the top row  
16 *             for (int i = left; i <= right; i++) {  
17 *                 result.add(mat[top][i]);  
18 *             }  
19 *             top++;  
20 *  
21 *             // 2. Traverse from top to bottom down the right column  
22 *             for (int i = top; i <= bottom; i++) {  
23 *                 result.add(mat[i][right]);  
24 *             }  
25 *             right--;  
26 *  
27 *             // 3. Traverse from right to left across the bottom row  
28 *             if (top <= bottom) {  
29 *                 for (int i = right; i >= left; i--) {  
30 *                     result.add(mat[bottom][i]);  
31 *                 }  
32 *                 bottom--;  
33 *             }  
34 *  
35 *             // 4. Traverse from bottom to top up the left column  
36 *             if (left <= right) {  
37 *                 for (int i = bottom; i >= top; i--) {  
38 *                     result.add(mat[i][left]);  
39 *                 }  
40 *             }  
41 *         }  
42 *     }  
43 * }
```

Output Window

Compilation Results Custom Input Y.O.G.I. (AI Bot)

Problem Solved Successfully ✓ Suggest Feedback

Test Cases Passed Attempts : Correct / Total 1 / 1 Accuracy : 100%

Custom Input Compile & Run Submit

https://leetcode.com/problems/search-a-2d-matrix/submissions/1909712494/ A ⭐ ... Chat

Problem List | Submit | 0 | 132:19:46 | Premium

Description | Editorial | Solutions | Submissions | Accepted | All Submissions

Code | Java

```
1 class Solution {  
2     public boolean searchMatrix(int[][] matrix, int target) {  
3         if (matrix == null || matrix.length == 0 || matrix[0].length == 0) {  
4             return false;  
5         }  
6  
7         int m = matrix.length;  
8         int n = matrix[0].length;  
9     }  
10    int mid = left + (right - left) / 2;  
11    int midValue = matrix[mid / n][mid % n];  
12  
13    if (midValue == target) {  
14        return true;  
15    } else if (midValue < target) {  
16        left = mid + 1;  
17    } else {  
18        right = mid - 1;  
19    }  
20}
```

More challenges

- 240. Search a 2D Matrix II
- 2468. Split Message Based on Limit

Write your notes here

Select related tags 0/5

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

https://www.geeksforgeeks.org/problems/median-in-a-row-wise-sorted-matrix1527/1

Get 90% Refund
Courses Tutorials Practice Jobs

Search... Problem Editorial Submissions Comments

Output Window

Compilation Results Custom Input Y.O.G.I. (AI Bot)

Problem Solved Successfully ✓ Suggest Feedback

Test Cases Passed 1117 / 1117 Attempts : Correct / Total 1 / 1 Accuracy : 100%

Points Scored 4 / 4 Time Taken 1.03

Your Total Score: 63 ↑

Solve Next

Reverse Spiral Form of Matrix Binary Matrix with at most K 1s

Aggressive Cows

Stay Ahead With:

Java (21) Start Timer

```
1 class Solution {  
2     public int median(int[][] mat) {  
3         // code here  
4         int n = mat.length;  
5         int m = mat[0].length;  
6         int min = Integer.MAX_VALUE;  
7         int max = Integer.MIN_VALUE;  
8  
9         for (int i = 0; i < n; i++) {  
10             min = Math.min(min, mat[i][0]);  
11             max = Math.max(max, mat[i][m - 1]);  
12         }  
13  
14         int desiredCount = (n * m + 1) / 2;  
15  
16         while (min < max) {  
17             int mid = min + (max - min) / 2;  
18             int count = 0;  
19  
20             for (int i = 0; i < n; i++) {  
21                 count += countSmallerEqual(mat[i], mid);  
22             }  
23  
24             if (count < desiredCount) {  
25                 min = mid + 1;  
26             } else {  
27                 max = mid;  
28             }  
29         }  
30         return min;  
31     }  
32  
33     private int countSmallerEqual(int[] row, int x) {  
34         int low = 0, high = row.length - 1;  
35         while (low <= high) {  
36             int mid = low + (high - low) / 2;  
37             if (row[mid] <= x) {  
38                 low = mid + 1;  
39             } else {  
40                 high = mid - 1;  
41             }  
42         }  
43         return low;  
44     }  
45 }
```

Custom Input Compile & Run Submit

https://www.geeksforgeeks.org/problems/row-with-max-1s0023/1

Get 90% Refund Courses Tutorials Practice Jobs

Search... Problem Editorial Submissions Comments

Difficulty: Medium Accuracy: 33.09% Submissions: 376K+ Points: 4

You are given a 2D binary array `arr[][]` consisting of only 1s and 0s. Each row of the array is sorted in non-decreasing order. Your task is to find and return the index of the first row that contains the maximum number of 1s. If no such row exists, return -1.

Note:

- The array follows 0-based indexing.
- The number of rows and columns in the array are denoted by n and m respectively.

Examples:

Input: arr[][] = [[0,1,1,1], [0,0,1,1], [1,1,1,1], [0,0,0,0]]
Output: 2

Output Window

Compilation Results Custom Input Y.O.G.I. (AI Bot)

Problem Solved Successfully ✓

Suggest Feedback

Test Cases Passed
1111 / 1111

Attempts : Correct / Total
1 / 1

Accuracy : 100%

Java (21) Start Timer

```
1 class Solution {
2     public int rowWithMax1s(int arr[][]) {
3         int n = arr.length;
4         int m = arr[0].length;
5         int maxRowIndex = -1;
6         int j = m - 1;
7
8         for (int i = 0; i < n; i++) {
9             while (j >= 0 && arr[i][j] == 1) {
10                 j--;
11                 maxRowIndex = i;
12             }
13         }
14
15         return maxRowIndex;
16     }
17 }
18 }
```

Custom Input Compile & Run Submit