

Description | Accepted | Editorial | Solutions | Submissions

< All Submissions

Accepted 26 / 26 testcases passed

Shreya submitted at Sep 16, 2025 20:50

Editorial

Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

41.32 MB | Beats 93.21%



Code | Java

```
class Solution {  
    public List<Integer> spiralOrder(int[][] matrix) {
```

Code

Java | Auto

```
1 class Solution {  
2     public List<Integer> spiralOrder(int[][] matrix) {  
3         List<Integer> result = new ArrayList<>();  
4         if (matrix == null || matrix.length == 0) return result;  
5  
6         int top = 0, bottom = matrix.length - 1;  
7         int left = 0, right = matrix[0].length - 1;  
8  
9         while (top <= bottom && left <= right) {  
10             // Traverse from Left to Right  
11             for (int j = left; j <= right; j++)  
12                 result.add(matrix[top][j]);  
13             top++;  
14  
15             // Traverse from Top to Bottom  
16             for (int i = top; i <= bottom; i++)  
17                 result.add(matrix[i][right]);  
18             right--;  
19  
20             // Traverse from Right to Left  
21             if (top <= bottom) {  
22                 for (int j = right; j >= left; j--)  
23                     result.add(matrix[bottom][j]);  
24                 bottom--;  
25             }  
26         }  
27     }  
28 }
```

Saved

Ln 37, Col 2

Testcase | Test Result

Accepted 21 / 21 testcases passed  
 Shreya submitted at Sep 16, 2025 20:55

Editorial

Solution

### Runtime

0 ms | Beats 100.00%

Analyze Complexity

### Memory

42.15 MB | Beats 78.56%



Code | Java

```
class Solution {
    public void rotate(int[][] matrix) {
        int n = matrix.length;
```

### Code

Java Auto

```
1 class Solution {
2     public void rotate(int[][] matrix) {
3         int n = matrix.length;
4
5         // Step 1: Transpose the matrix
6         for (int i = 0; i < n; i++) {
7             for (int j = i + 1; j < n; j++) {
8                 int temp = matrix[i][j];
9                 matrix[i][j] = matrix[j][i];
10                matrix[j][i] = temp;
11            }
12        }
13    }
14 }
```

Saved

Ln 23, Col 2

### Testcase Test Result

[[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]]


### Expected

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

Contribute a testcase

 Search...


Courses ▼Tutorials ▼Practice ▼Jobs ▼



ProblemEditorialSubmissionsComments



Output Window

Compilation ResultsCustom InputY.O.G.I. (AI Bot)

Problem Solved Successfully  [Suggest Feedback](#)

Test Cases Passed  
**1120 / 1120**

Attempts : Correct / Total  
**1 / 1**  
Accuracy : 100%

Points Scored   
**4 / 4**  
Your Total Score: 4 

Time Taken  
**1.73**

Solve Next

Rotate by 90 degreeSummed MatrixRotate matrix elements clockwise

Java (21)Start Timer

```
1 class Solution {
2     public void rotateMatrix(int[][] matrix) {
3         int n = matrix.length;
4
5         // Reverse both rows and columns
6         for (int i = 0; i < n / 2; i++) {
7             for (int j = 0; j < n; j++) {
8                 int temp = matrix[i][j];
9                 matrix[i][j] = matrix[n - i - 1][n - j - 1];
10                matrix[n - i - 1][n - j - 1] = temp;
11            }
12        }
13
14        // If n is odd, reverse the middle row
15        if (n % 2 == 1) {
16            int mid = n / 2;
17            for (int j = 0; j < n / 2; j++) {
18                int temp = matrix[mid][j];
19                matrix[mid][j] = matrix[mid][n - j - 1];
20                matrix[mid][n - j - 1] = temp;
21            }
22        }
23    }
24 }
```

Custom InputCompile & RunSubmit