Problem | Editorial | Submissions

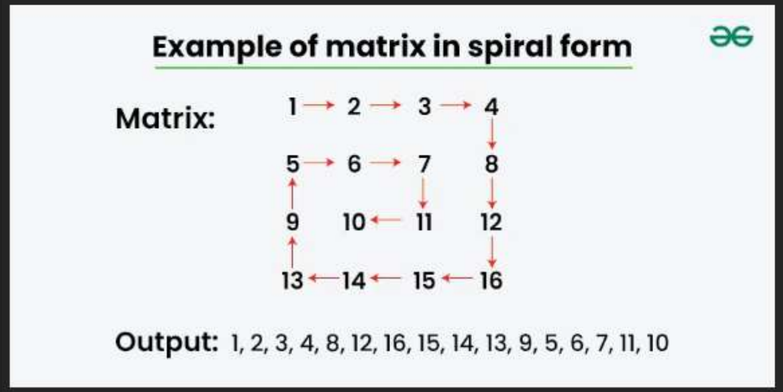
Spirally traversing a matrix

Difficulty: **Medium** Accuracy: **35.2%** Submissions: **303K+** Points: **4**

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:



Input: mat[][] = [[1, 2, 3, 4, 5, 6], [7, 8, 9, 10, 11, 12], [13, 14, 15, 16, 17,

Java (1.8) | Start Timer

```java
// } Driver Code Ends

class Solution {
    public ArrayList<Integer> spirallyTraverse(int mat[][]) {
        ArrayList<Integer> result = new ArrayList<>();
        if (mat == null || mat.length == 0) return result;
        int top = 0;
        int bottom = mat.length - 1;
        int left = 0;
        int right = mat[0].length - 1;
        while (top <= bottom && left <= right) {
            for (int i = left; i <= right; i++) {
                result.add(mat[top][i]);
            }
            top++;
            for (int i = top; i <= bottom; i++) {
                result.add(mat[i][right]);
            }
            right--;
            if (top <= bottom) {
                for (int i = right; i >= left; i--) {
                    result.add(mat[bottom][i]);
                }
                bottom--;
            }
            if (left <= right) {
                for (int i = bottom; i >= top; i--) {
                    result.add(mat[i][left]);
                }
                left++;
            }
        }
        return result;
    }
}
```

Custom Input | Compile & Run | Submit