Day-51: Spiral Matrix (LeetCode 54)

Problem Statement

Given a 2D matrix, return all elements of the matrix in spiral order (clockwise).

Example

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

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

Approach

Use four boundaries (top, bottom, left, right) and traverse the matrix layer by layer in spiral order while shrinking the boundaries.

Dry Run Explanation

```
top=0, bottom=2, left=0, right=2

Top row    -> 1 2 3
Right col  -> 6 9
Bottom row -> 8 7
Left col   -> 4
Center     -> 5
```

Java Code

```
import java.util.*;

class SpiralMatrix {
    public static List<Integer> spiralOrder(int[][] matrix) {
        List<Integer> result = new ArrayList<>();
        int top = 0, bottom = matrix.length - 1;
        int left = 0, right = matrix[0].length - 1;

        while (top <= bottom && left <= right) {
            for (int i = left; i <= right; i++)
                result.add(matrix[top][i]);
            top++;

            for (int i = top; i <= bottom; i++)
                result.add(matrix[i][right]);
            right--;

            if (top <= bottom) {
                for (int i = right; i >= left; i--)
                    result.add(matrix[bottom][i]);
                bottom--;
            }

            if (left <= right) {
                for (int i = bottom; i >= top; i--)
                    result.add(matrix[i][left]);
                left++;
            }
        }
        return result;
```

```
        }
    }
```

C++ Code

```
vector<int> spiralOrder(vector<vector<int>>& matrix) {
    vector<int> res;
    int top=0, bottom=matrix.size()-1;
    int left=0, right=matrix[0].size()-1;

    while(top<=bottom && left<=right){
        for(int i=left;i<=right;i++) res.push_back(matrix[top][i]);
        top++;
        for(int i=top;i<=bottom;i++) res.push_back(matrix[i][right]);
        right--;
        if(top<=bottom){
            for(int i=right;i>=left;i--) res.push_back(matrix[bottom][i]);
            bottom--;
        }
        if(left<=right){
            for(int i=bottom;i>=top;i--) res.push_back(matrix[i][left]);
            left++;
        }
    }
    return res;
}
```

Python Code

```
def spiralOrder(matrix):
    res=[]
    top,bottom=0,len(matrix)-1
    left,right=0,len(matrix[0])-1

    while top<=bottom and left<=right:
        for i in range(left,right+1):
            res.append(matrix[top][i])
        top+=1
        for i in range(top,bottom+1):
            res.append(matrix[i][right])
        right-=1
        if top<=bottom:
            for i in range(right,left-1,-1):
                res.append(matrix[bottom][i])
            bottom-=1
        if left<=right:
            for i in range(bottom,top-1,-1):
                res.append(matrix[i][left])
            left+=1
    return res
```

Time & Space Complexity

Time Complexity: O(m × n)
Space Complexity: O(1) (excluding output list)