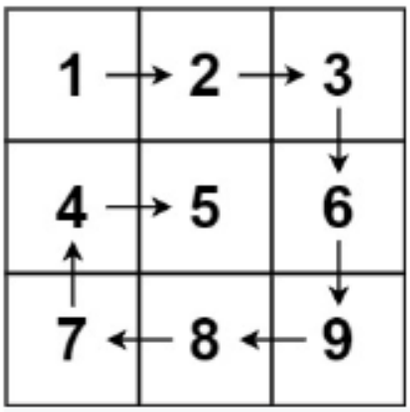
Spiral matrix

54. Spiral Matrix

Medium 12546 1120 Add to List Share

Given an `m x n` `matrix`, return *all elements of the* `matrix` *in spiral order.*

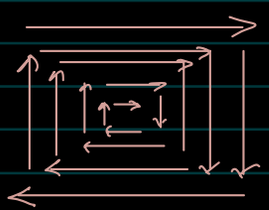
Example 1:




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

In this problem we have to traverse matrix in spiral format

i.e



we will required one loop in one direction i.e in total 4 loops

From Top

1st loop: left to Right →

2nd loop: top to Bottom ↓

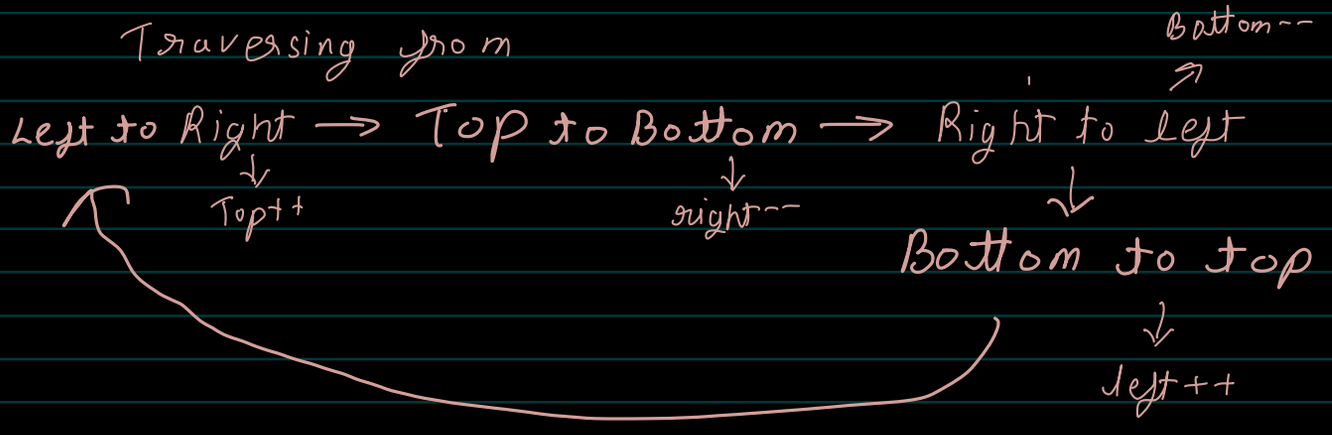
3rd loop: Right to left from Bottom ←

4th loop: bottom to top-1

left ↓ 0 1 2 right ↓ 3

Top - 0
1
2
Bottom → 3

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



1st loop:
for (i: left to right)
a[top][i]
top++

2nd loop:
for (i: top to Bottom)
a[i][right]

3rd loop: right--
if (top <= Bottom)
for (i: Right → Left)
a[i][bottom]

4th loop: Bottom--
if (left <= right)
for (i: Bottom → top)
a[left][i]
left++

Code

```cpp
class Solution {
public:
    vector<int> spiralOrder(vector<vector<int>>& matrix) {
        int n=matrix.size();
        int m=matrix[0].size();
        int top=0,left=0,right=m-1,bottom=n-1;
        vector<int> ans;
        while(top <=bottom && left<=right){
            for(int i=left;i<=right;i++){
                ans.push_back(matrix[top][i]);
            }
            top++;
            for(int i=top;i<=bottom;i++){
                ans.push_back(matrix[i][right]);
            }
            right--;
            if(top<=bottom){
                for(int i=right;i>=left;i--){
                    ans.push_back(matrix[bottom][i]);
                }
            }
            bottom--;
            if(left<=right){
                for(int i=bottom;i>=top;i--){
                    ans.push_back(matrix[i][left]);
                }
            }
            left++;
        }
        return ans;
    }
};
```

T.C - O(n x m)

S.C - O(n) → storing in ans