

Description

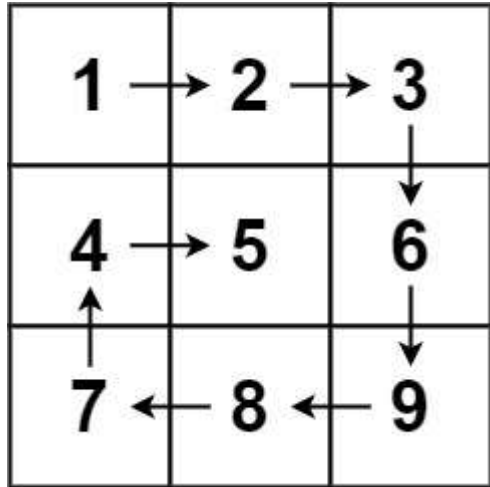
Solution

Discuss (999+)

Submissions

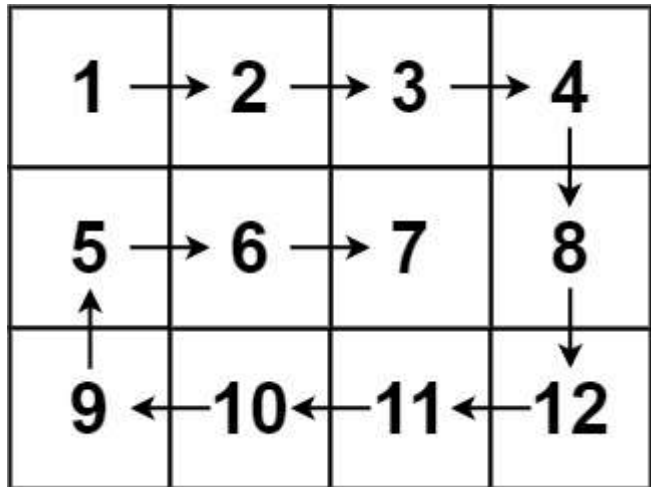
Given an $m \times n$ matrix, return *all elements of the matrix in spiral order*.

Example 1:



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

Example 2:



Input: matrix = [[1,2,3,4],[5,6,7,8],[9,10,11,12]]
Output: [1,2,3,4,8,12,11,10,9,5,6,7]

Constraints:

- $m == \text{matrix.length}$

Java

Autocomplete

i

{ }

↶

⚙

⌕

```
1  class Solution {
2
3  public List<Integer> spiralOrder(int[][] matrix) {
4
5      int m = matrix.length, n = matrix[0].length;
6      List<Integer> list = new LinkedList<>();
7
8      int left = 0, top = 0;
9      int right = n - 1, bottom = m - 1;
10
11     while(true){
12
13         for(int i = left; i <= right; i++){
14             list.add(matrix[top][i]);
15         }
16
17         top++;
18
19         if(left > right || top > bottom)
20             break;
21
22         for(int i = top; i <= bottom; i++){
23             list.add(matrix[i][right]);
24         }
25
26         right--;
27
28         if(left > right || top > bottom)
29             break;
30
31         for(int i = right; i >= left; i--){
32             list.add(matrix[bottom][i]);
33         }
34
35         bottom--;
36
37         if(left > right || top > bottom)
38             break;
39
40         for(int i = bottom; i >= top; i--){
41             list.add(matrix[i][left]);
42         }
43
44         left++;
45
46         if(left > right || top > bottom)
47             break;
48     }
49     return list;
}
```