1. 题目
2. 题干

给定一个包含 m x n 个元素的矩阵（m 行, n 列），请按照顺时针螺旋顺序，返回矩阵中的所有元素。

1. 示例

示例 1:

输入:

[

[ 1, 2, 3 ],

[ 4, 5, 6 ],

[ 7, 8, 9 ]

]

输出: [1,2,3,6,9,8,7,4,5]

示例 2:

输入:

[

[1, 2, 3, 4],

[5, 6, 7, 8],

[9,10,11,12]

]

输出: [1,2,3,4,8,12,11,10,9,5,6,7]

1. 题解
2. 思路

又是一道数组的模拟题目，最简单的方法就是按照顺时针的顺序模拟一遍，总共4个方向，右，下，左，上。在每个拐角处转变方向，注意不同的拐角处处理方式不同，且螺旋的范围是不断缩小的，这一点是重点。

1. 代码实现

Java：

class Solution {

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

        List<Integer> list = new ArrayList<>();

        if(matrix==null || matrix.length == 0){

            return list;

        }

        int mIndex = matrix.length-1;

        int nIndex = matrix[0].length-1;

        int m = mIndex; //变化的最大行索引

        int n = nIndex; //变化的最大列索引

        int direction = 1;

        int i = 0, j = 0;   //i行坐标，j纵坐标

        for(int k = 0; k < matrix.length \* matrix[0].length; ++k){

            list.add(matrix[i][j]);

            if(direction == 1){

                if(j == n){

                    direction = 2;

                    i++;

                }else{

                    j++;

                }

                continue;

            }

            if(direction == 2){

                if(i == m){

                    direction = 3;

                    j--;

                }else{

                    i++;

                }

                continue;

            }

            if(direction == 3){

                if(j == nIndex-n){

                    direction =4;

                    m--;

                    n--;

                    i--;

                }else{

                    j--;

                }

                continue;

            }

            if(direction == 4){

                if(i == mIndex - m){

                    direction = 1;

                    j++;

                }else{

                    i--;

                }

                continue;

            }

        }

        return list;

    }

}