

0054. 螺旋矩阵

👤 ITCharge ⌚ 大约 1 分钟

- 标签：数组、矩阵、模拟
- 难度：中等

题目链接

- [0054. 螺旋矩阵 - 力扣](#)

题目大意

描述： 给定一个 $m \times n$ 大小的二维矩阵 $matrix$ 。

要求： 按照顺时针旋转的顺序，返回矩阵中的所有元素。

说明：

- $m == matrix.length$ 。
- $n == matrix[i].length$ 。
- $1 \leq m, n \leq 10$ 。
- $-100 \leq matrix[i][j] \leq 100$ 。

示例：

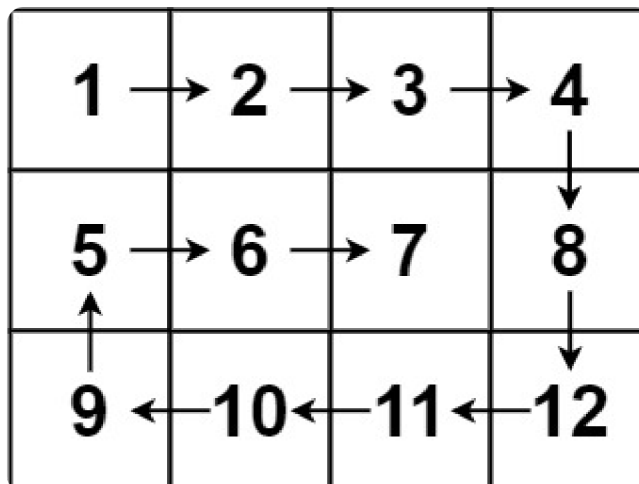
- 示例 1：

1	→ 2	→ 3
4	→ 5	↓ 6
↑ 7	← 8	← 9

py

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

• 示例 2:



py

输入: `matrix = [[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]`

解题思路

思路 1: 模拟

1. 使用数组 `ans` 存储答案。然后定义一下上、下、左、右的边界。
2. 然后按照逆时针的顺序从边界上依次访问元素。
3. 当访问完当前边界之后，要更新一下边界位置，缩小范围，方便下一轮进行访问。
4. 最后返回答案数组 `ans`。

思路 1: 代码

py

```
class Solution:
    def spiralOrder(self, matrix: List[List[int]]) -> List[int]:
        up, down, left, right = 0, len(matrix)-1, 0, len(matrix[0])-1
        ans = []
        while True:
            for i in range(left, right + 1):
```

```
        ans.append(matrix[up][i])
    up += 1
    if up > down:
        break
    for i in range(up, down + 1):
        ans.append(matrix[i][right])
    right -= 1
    if right < left:
        break
    for i in range(right, left - 1, -1):
        ans.append(matrix[down][i])
    down -= 1
    if down < up:
        break
    for i in range(down, up - 1, -1):
        ans.append(matrix[i][left])
    left += 1
    if left > right:
        break
return ans
```

思路 1：复杂度分析

- **时间复杂度：** $O(m \times n)$ 。其中 m 、 n 分别为二维矩阵的行数和列数。
- **空间复杂度：** $O(m \times n)$ 。如果算上答案数组的空间占用，则空间复杂度为 $O(m \times n)$ 。不算上则空间复杂度为 $O(1)$ 。