

0059. 螺旋矩阵 II

👤 [ITCharge](#) ⌚ 大约 1 分钟

- 标签：数组、矩阵、模拟
- 难度：中等

题目链接

- [0059. 螺旋矩阵 II - 力扣](#)

题目大意

给你一个正整数 n 。

要求：生成一个包含 $1 \sim n^2$ 的所有元素，且元素按顺时针顺序螺旋排列的 $n \times n$ 正方形矩阵 $matrix$ 。

解题思路

思路 1：模拟

这道题跟「[54. 螺旋矩阵](#)」思路是一样的。

1. 构建一个 $n \times n$ 大小的数组 $matrix$ 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给数组 $matrix$ 相应位置赋值。
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 $matrix$ 。

思路 1：代码

py

```
class Solution:
    def generateMatrix(self, n: int) -> List[List[int]]:
        matrix = [[0 for _ in range(n)] for _ in range(n)]
        up, down, left, right = 0, len(matrix) - 1, 0, len(matrix[0]) - 1
        index = 1
        while True:
            for i in range(left, right + 1):
                matrix[up][i] = index
                index += 1
            up += 1
            if up > down:
                break
            for i in range(up, down + 1):
                matrix[i][right] = index
                index += 1
            right -= 1
            if right < left:
                break
            for i in range(right, left - 1, -1):
                matrix[down][i] = index
                index += 1
            down -= 1
            if down < up:
                break
            for i in range(down, up - 1, -1):
                matrix[i][left] = index
                index += 1
            left += 1
            if left > right:
                break
        return matrix
```

思路 1：复杂度分析

- 时间复杂度： $O(n^2)$ 。

- 空间复杂度: $O(n^2)$ 。