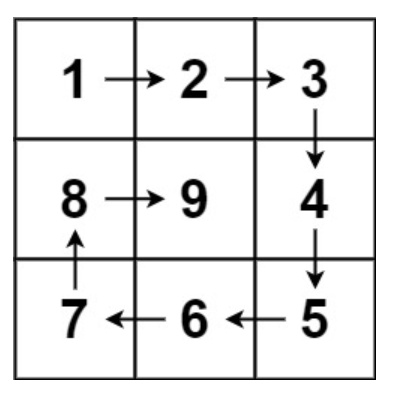
# 题目

给你一个正整数n，生成一个包含1到n2所有元素，且元素按顺时针顺序螺旋排列的n x n正方形矩阵matrix。

示例 1：



输入：n = 3

输出：[[1,2,3],[8,9,4],[7,6,5]]

示例 2：

输入：n = 1

输出：[[1]]

提示：

1 <= n <= 20

# 分析

## 方法一：模拟（推荐）

**解题思路：**

1、初始化矩阵和边界：

创建一个 𝑛×𝑛的矩阵 matrix，初始化为 0。

定义四个边界：top = 0, bottom = n - 1, left = 0, right = n - 1。

2、顺时针填充：

1）按照螺旋顺序依次填充：

从左到右填充当前上边界。

从上到下填充当前右边界。

从右到左填充当前下边界（如果下边界有效）。

从下到上填充当前左边界（如果左边界有效）。

2）每次填充完一个方向后，更新对应的边界。

3、终止条件：

当所有元素都被填充完，即当前数值 num > n^2 时停止。

代码：

class Solution {

public:

vector<vector<int>> generateMatrix(int n) {

vector<vector<int>> matrix(n, vector<int>(n, 0)); // 初始化 n x n 矩阵

int top = 0, bottom = n - 1, left = 0, right = n - 1;

int num = 1, maxNum = n \* n;

while (num <= maxNum) {

// 1. 从左到右填充当前上边界

for (int j = left; j <= right && num <= maxNum; ++j) {

matrix[top][j] = num++;

}

++top; // 收缩上边界

// 2. 从上到下填充当前右边界

for (int i = top; i <= bottom && num <= maxNum; ++i) {

matrix[i][right] = num++;

}

--right; // 收缩右边界

// 3. 从右到左填充当前下边界

for (int j = right; j >= left && num <= maxNum; --j) {

matrix[bottom][j] = num++;

}

--bottom; // 收缩下边界

// 4. 从下到上填充当前左边界

for (int i = bottom; i >= top && num <= maxNum; --i) {

matrix[i][left] = num++;

}

++left; // 收缩左边界

}

return matrix;

}

};

时间复杂度：

O(n2)：每个元素访问一次。

空间复杂度：

O(1)：仅使用常数额外空间（不计结果矩阵）。

## 方法一：模拟

思路：

使用模拟的方法来生成螺旋矩阵。定义四个变量 top, bottom, left, right 分别表示当前要填充的矩阵的上边界、下边界、左边界和右边界。然后，从左到右、从上到下、从右到左、从下到上依次填充矩阵，每填充完一行或一列就更新对应的边界。

代码：

class Solution {

public:

vector<vector<int>> generateMatrix(int n) {

vector<vector<int>> matrix(n, vector<int>(n, 0));

int num = 1;

int top = 0, bottom = n - 1, left = 0, right = n - 1;

while (num <= n \* n) {

// 从左到右填充上边界

for (int j = left; j <= right; ++j) {

matrix[top][j] = num++;

}

++top;

// 从上到下填充右边界

for (int i = top; i <= bottom; ++i) {

matrix[i][right] = num++;

}

--right;

// 从右到左填充下边界

for (int j = right; j >= left; --j) {

matrix[bottom][j] = num++;

}

--bottom;

// 从下到上填充左边界

for (int i = bottom; i >= top; --i) {

matrix[i][left] = num++;

}

++left;

}

return matrix;

}

};