

5月4日报

本日学习内容

- 1. 学习面向对象下的类别与拓展，用类别对类进行模块化设计，用类别来调用私有方法
- 2. 完成hot 100矩阵部分

今日算法题

题目1: [54. 螺旋矩阵](#)

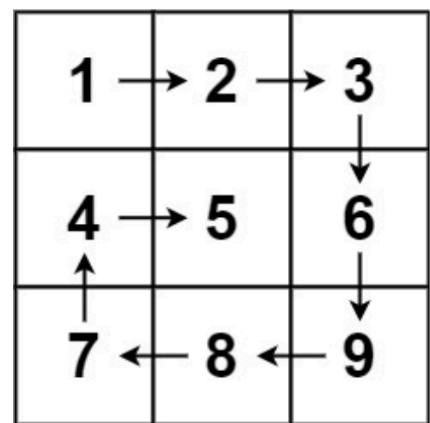
54. 螺旋矩阵

已解答

中等 相关标签 相关企业 提示 Aa

给你一个 m 行 n 列的矩阵 `matrix`，请按照 顺时针螺旋顺序，返回矩阵中的所有元素。

示例 1:



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

```
class Solution {
public:
    vector<int> spiralOrder(vector<vector<int>>& matrix) {
        vector<int> result;
        if (matrix.empty() || matrix[0].empty())
            return result;
        int m = matrix.size();
        int n = matrix[0].size();
        int top = 0, bottom = m - 1;
        int left = 0, right = n - 1;
        while (top <= bottom && left <= right) {
            for (int j = left; j <= right; ++j) {
                result.push_back(matrix[top][j]);
            }
            top++;
            for (int i = top; i <= bottom; ++i) {
                result.push_back(matrix[i][right]);
            }
            right--;
            if (left <= right) {
                for (int j = right; j >= left; --j) {
                    result.push_back(matrix[bottom][j]);
                }
                bottom--;
            }
            if (top <= bottom) {
                for (int i = bottom; i >= top; --i) {
                    result.push_back(matrix[i][left]);
                }
                left++;
            }
        }
        return result;
    }
};
```

```

    }
    top++;
    for (int i = top; i <= bottom; ++i) {
        result.push_back(matrix[i][right]);
    }
    right--;
    if (top <= bottom) {
        for (int j = right; j >= left; --j) {
            result.push_back(matrix[bottom][j]);
        }
        bottom--;
    }
    if (left <= right) {
        for (int i = bottom; i >= top; --i) {
            result.push_back(matrix[i][left]);
        }
        left++;
    }
}
return result;
}
};

```

题目2: [73. 矩阵置零](#)

73. 矩阵置零

已解答 ✓

中等

🔖 相关标签

🔒 相关企业

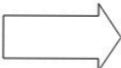
💡 提示

Ax

给定一个 $m \times n$ 的矩阵，如果一个元素为 0，则将其所在行和列的所有元素都设为 0。请使用 [原地](#) 算法。

示例 1:

1	1	1
1	0	1
1	1	1




1	0	1
0	0	0
1	0	1

输入: matrix = [[1,1,1],[1,0,1],[1,1,1]]

输出: [[1,0,1],[0,0,0],[1,0,1]]

示例 2:

0	1	2	0
3	4	5	2
1	3	1	5



0	0	0	0
0	4	5	0
0	3	1	0

输入: matrix = [[0,1,2,0],[3,4,5,2],[1,3,1,5]]

输出: [[0,0,0,0],[0,4,5,0],[0,3,1,0]]

```
class Solution {
public:
    void setZeroes(vector<vector<int>>& matrix) {
        int m = matrix.size();
        int n = matrix[0].size();
        vector<int> row(m), col(n);
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                if (!matrix[i][j]) {
                    row[i] = col[j] = true;
                }
            }
        }
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                if (row[i] || col[j]) {
                    matrix[i][j] = 0;
                }
            }
        }
    }
}
```

```
    }  
}  
};
```

本日遇到的问题

1. 初学时不太清楚拓展的定义，导致与类有些区分不清
2. oc代码能力还需要加强，不能独立完成用代码实现拓展类

明日学习计划

1. 开始学习面向对象下的协议与委托，使用类别实现非正式协议，遵守实现协议
2. 开始复习回溯算法，在leetcode完成相关题目