

第 8 章

数组和矩阵问题

转圈打印矩阵

【题目】

给定一个整型矩阵 `matrix`，请按照转圈的方式打印它。

例如：

1 2 3 4

5 6 7 8

9 10 11 12

13 14 15 16

打印结果为：1, 2, 3, 4, 8, 12, 16, 15, 14, 13, 9, 5, 6, 7, 11, 10

【要求】

额外空间复杂度为 $O(1)$ 。

【难度】

士 ★☆☆☆

【解答】

本题在算法上没有难度，关键在于设计一种逻辑容易理解、代码易于实现的转圈遍历

方式。这里介绍这样一种矩阵处理方式，该方式不仅可用于这道题，还适合很多其他的面试题，就是矩阵分圈处理。在矩阵中用左上角的坐标(tR, tC)和右下角的坐标(dR, dC)就可以表示一个子矩阵，比如，题目中的矩阵，当(tR, tC)=(0,0)、(dR, dC)=(3,3)时，表示的子矩阵就是整个矩阵，那么这个子矩阵最外层的部分如下：

```
1    2    3    4
5                8
9                12
13   14   15   16
```

如果能把这个子矩阵的外层转圈打印出来，那么在(tR, tC)=(0,0)、(dR, dC)=(3,3)时，打印的结果为：1, 2, 3, 4, 8, 12, 16, 15, 14, 13, 9, 5。接下来令 tR 和 tC 加 1，即(tR, tC)=(1,1)，令 dR 和 dC 减 1，即(dR, dC)=(2,2)，此时表示的子矩阵如下：

```
6    7
10   11
```

再把这个子矩阵转圈打印出来，结果为：6, 7, 11, 10。把 tR 和 tC 加 1，即(tR, tC)=(2,2)，令 dR 和 dC 减 1，即(dR, dC)=(1,1)。如果发现左上角坐标跑到了右下角坐标的右方或下方，整个过程就停止。已经打印的所有结果连起来就是我们要求的打印结果。具体请参看如下代码中的 `spiralOrderPrint` 方法，其中 `printEdge` 方法是转圈打印一个子矩阵的外层。

```
public void spiralOrderPrint(int[][] matrix) {
    int tR = 0;
    int tC = 0;
    int dR = matrix.length - 1;
    int dC = matrix[0].length - 1;
    while (tR <= dR && tC <= dC) {
        printEdge(matrix, tR++, tC++, dR--, dC--);
    }
}

public void printEdge(int[][] m, int tR, int tC, int dR, int dC) {
    if (tR == dR) { // 子矩阵只有一行时
        for (int i = tC; i <= dC; i++) {
            System.out.print(m[tR][i] + " ");
        }
    } else if (tC == dC) { // 子矩阵只有一列时
        for (int i = tR; i <= dR; i++) {
            System.out.print(m[i][tC] + " ");
        }
    } else { // 一般情况
        int curC = tC;
        int curR = tR;
```

```

        while (curC != dC) {
            System.out.print(m[tR][curC] + " ");
            curC++;
        }
        while (curR != dR) {
            System.out.print(m[curR][dC] + " ");
            curR++;
        }
        while (curC != tC) {
            System.out.print(m[dR][curC] + " ");
            curC--;
        }
        while (curR != tR) {
            System.out.print(m[curR][tC] + " ");
            curR--;
        }
    }
}

```

将正方形矩阵顺时针转动 90°

【题目】

给定一个 $N \times N$ 的矩阵 `matrix`，把这个矩阵调整成顺时针转动 90° 后的形式。

例如：

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

顺时针转动 90° 后为：

13	9	5	1
14	10	6	2
15	11	7	3
16	12	8	4

【要求】

额外空间复杂度为 $O(1)$ 。