

P4924 [1007]魔法少女小Scarlet

题目描述

Scarlet最近学会了一个数组魔法，她会在 $n \times n$ 二维数组上将一个奇数阶方阵按照顺时针或者逆时针旋转 90° ，

首先，Scarlet会把1到 n^2 的正整数按照从左往右，从上至下的顺序填入初始的二维数组中，然后她会施放一些简易的魔法。

Scarlet既不会什么分块特技，也不会什么Splay套Splay，她现在提供给你她的魔法执行顺序，想让你来告诉她魔法按次执行完毕后的二维数组。

输入格式

第一行两个整数 n, m ，表示方阵大小和魔法施放次数。

接下来 m 行，每行4个整数 x, y, r, z 表示在这次魔法中，Scarlet会把以第 x 行第 y 列为中心的 $2r+1$ 阶矩阵按照某种时针方向旋转，其中 $z=0$ 表示顺时针， $z=1$ 表示逆时针。

输出格式

输出 n 行，每行 n 个用空格隔开的数，表示最终所得的矩阵

输入样例

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

输出样例

```
5 10 3 18 15
4 19 8 17 20
1 14 23 24 25
6 9 2 7 22
11 12 13 16 21
```

解析

先搞懂题意，首先是边长为 n 的矩阵，题目中给的数据是5，则如下图所示：

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

21	22	23	24	25
----	----	----	----	----

2 2 1 0: 以第2行，第2列为中心的3阶矩阵按照顺时针进行旋转。

11	6	1	4	5
12	7	2	9	10
13	8	3	14	15
16	17	18	19	20
21	22	23	24	25

3 3 1 1: 以第3行，第3列为中心的3阶矩阵按照逆时针进行旋转。

11	6	1	4	5
12	9	14	19	10
13	2	3	18	15
16	7	8	17	20
21	22	23	24	25

4 4 1 0: 以第4行，第4列为中心的3阶矩阵按照顺时针进行旋转。

11	6	1	4	5
12	9	14	19	10
13	2	23	8	3
16	7	24	17	18
21	22	25	20	15

3 3 2 1: 以第3行，第3列为中心的5阶矩阵按照逆时针进行旋转。

5	10	3	18	15
4	19	8	17	20
1	14	23	24	25
6	9	2	7	22
11	12	13	16	21

本题的关键就是旋转，如何却定旋转的范围以及旋转的变化过程是我们首要解决的问题。我们以3x3数组为例来尝试寻找规律。

	-1	0	1
-1	1	2	3
0	4	5	6
1	7	8	9

	-1	0	1
-1	7	4	1
0	8	5	2
1	9	6	3

数字1从 $\{-1, -1\}$ 变成了 $\{-1, 1\}$

数字2从 $\{-1, 0\}$ 变成了 $\{0, 1\}$

数字3从 $\{-1, 1\}$ 变成了 $\{1, 1\}$

即数字 n 从 $\{i, j\}$ 坐标变成 $\{j, -i\}$

那如果我们一开始中心点处于某个其他的坐标呢？例如在2, 3，如下图所示：

	2	3	4
1	1	2	3
2	4	5	6
3	7	8	9

	2	3	4
1	7	4	1
2	8	5	2
3	9	6	3

数字1从 $\{1(-1+2), 2(-1+3)\}$ 变成了 $\{1(-1+2), 4(1+3)\}$

数字2从 $\{1(-1+2), 3(0+3)\}$ 变成了 $\{2(0+2), 4(1+3)\}$

数字3从 $\{1(-1+2), 4(1+3)\}$ 变成了 $\{3(1+2), 4(1+3)\}$

发现规律了吧，变化规律依然是从 $\{i, j\}$ 坐标变成 $\{j, -i\}$ ，只不过要加上偏移的中心点坐标，即从 $\{i+y, j+x\}$ 坐标变成 $\{j+y, x-i\}$ ，用代码表示即为：

$$b[y + j][x - i] = a[y + i][x + j];$$

同理，我们也可以推出向左旋转90度的代码公式为： $\{i+y, j+x\} \Rightarrow \{y-j, x+i\}$ ，用代码表示即为：

$$b[y - j][x + i] = a[y + i][x + j];$$

因此，最终的旋转编码为：

```
//顺时针
void clockwise(int y, int x, int r) {
    //双重for循环定位中心点和范围
    for (int i = -r; i <= r; i++) {
        for (int j = -r; j <= r; j++) {
            b[y + j][x - i] = a[y + i][x + j];
        }
    }
}

//逆时针
```

```

void anticlockwise(int y, int x, int r) {
    //双重for循环定位中心点和范围
    for (int i = -r; i <= r; i++) {
        for (int j = -r; j <= r; j++) {
            b[y - j][x + i] = a[y + i][x + j];
        }
    }
}

```

编码

```

#include<bits/stdc++.h>

using namespace std;

//原始数组
int a[505][505];
//新数组
int b[505][505];
//矩阵边长和询问次数
int n, m;

void Out() {
    //输出最终的队列
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            printf("%d ", a[i][j]);
        }
        printf("\n");
    }
}

void init() {
    scanf("%d%d", &n, &m);
    //初始化数组，存入数据
    int v = 1;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            a[i][j] = v++;
        }
    }
}

//顺时针
void clockwise(int y, int x, int r) {
    //双重for循环定位中心点和范围
    for (int i = -r; i <= r; i++) {
        for (int j = -r; j <= r; j++) {

```

```

        b[y + j][x - i] = a[y + i][x + j];
    }
}

```

//逆时针

```

void anticlockwise(int y, int x, int r) {
    //双重for循环定位中心点和范围
    for (int i = -r; i <= r; i++) {
        for (int j = -r; j <= r; j++) {
            b[y - j][x + i] = a[y + i][x + j];
        }
    }
}

```

//旋转完毕之后将数据拷贝回原始数组

```

void CopyToOrg() {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            if (b[i][j] > 0) {
                a[i][j] = b[i][j];
            }
        }
    }
}

```

//读入旋转数据，并执行旋转

```

void work() {
    int x, y, r, z;
    for (int i = 1; i <= m; i++) {
        scanf("%d%d%d%d", &x, &y, &r, &z);
        if (!z) {
            clockwise(x, y, r);
        } else {
            anticlockwise(x, y, r);
        }
        CopyToOrg();
    }
    Out();
}

```

```

int main() {
    init();
    work();
    return 0;
}

```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。



