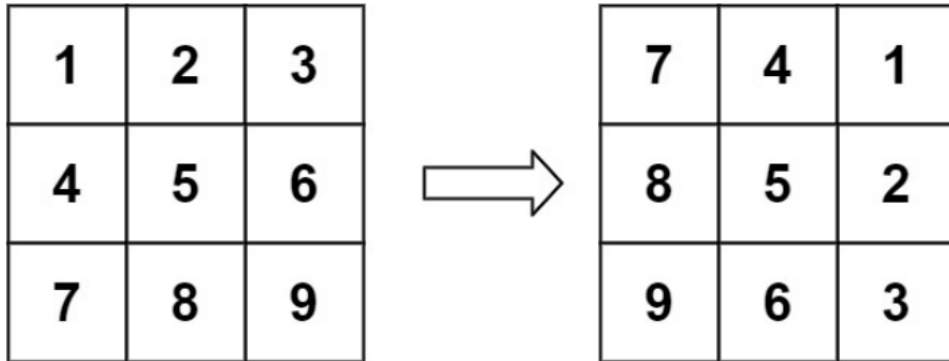


## 48 旋转图像

Label: 数组

给定一个  $n \times n$  的二维矩阵 `matrix` 表示一个图像。请你将图像顺时针旋转 90 度。

你必须在 原地 旋转图像，这意味着你需要直接修改输入的二维矩阵。请不要 使用另一个矩阵来旋转图像。



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

输出: `[[7,4,1],[8,5,2],[9,6,3]]`

- 原地遍历

```
class Solution {
    public static void rotate(int[][] matrix) {
        // 这两个指针表示最外层的边行和边列
        int abs1 = 0;
        int abs2 = matrix.length - 1;
        while (abs1 <= abs2) { // 控制内外圈, 逐层像内
            int p1 = abs1;
            int p2 = abs2;
            while (p1 != abs2) { // 一圈
                int temp = matrix[abs1][p1]; // 左上
                matrix[abs1][p1] = matrix[p2][abs1]; // 左上 = 左下
                matrix[p2][abs1] = matrix[abs2][p2]; // 左下 = 右下
                matrix[abs2][p2] = matrix[p1][abs2]; // 右下 = 右上
                matrix[p1][abs2] = temp; // 右上 = 左上

                p1 += 1;
                p2 -= 1;
            }
            abs1 += 1;
            abs2 -= 1;
        }
    }
}
```

- 辅助数组

```
class Solution {  
  
    public void rotate(int[][] matrix) {  
        int n = matrix.length;  
        int[][] matrix_new = new int[n][n];  
  
        for (int i = 0; i < n; ++i) {  
            for (int j = 0; j < n; ++j) {  
                matrix_new[j][n - i - 1] = matrix[i][j];  
            }  
        }  
  
        for (int i = 0; i < n; ++i) {  
            for (int j = 0; j < n; ++j) {  
                matrix[i][j] = matrix_new[i][j];  
            }  
        }  
    }  
}
```

- 递归

```
class Solution {  
  
    public void rotate(int[][] matrix) {  
        int len = matrix.length;  
        r(matrix, 0, len - 1, len - 1);  
    }  
  
    public void r(int[][] matrix, int start, int end, int num) {  
        if(start < end){  
            for(int i = 0; i < num; i++){  
                int tmp = matrix[start][start + i];  
                matrix[start][start + i] = matrix[end - i][start];  
                matrix[end - i][start] = matrix[end][end - i];  
                matrix[end][end - i] = matrix[start + i][end];  
                matrix[start + i][end] = tmp;  
            }  
            r(matrix, start + 1, end - 1, num - 2);  
        }  
    }  
}
```

- 镜像操作