# 题目

给定一个n × 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。

示例 1：



输入：matrix = [[1,2,3],[4,5,6],[7,8,9]]

输出：[[7,4,1],[8,5,2],[9,6,3]]

示例 2：



输入：matrix = [[5,1,9,11],[2,4,8,10],[13,3,6,7],[15,14,12,16]]

输出：[[15,13,2,5],[14,3,4,1],[12,6,8,9],[16,7,10,11]]

示例 3：

输入：matrix = [[1]]

输出：[[1]]

示例 4：

输入：matrix = [[1,2],[3,4]]

输出：[[3,1],[4,2]]

提示：

matrix.length == n

matrix[i].length == n

1 <= n <= 20

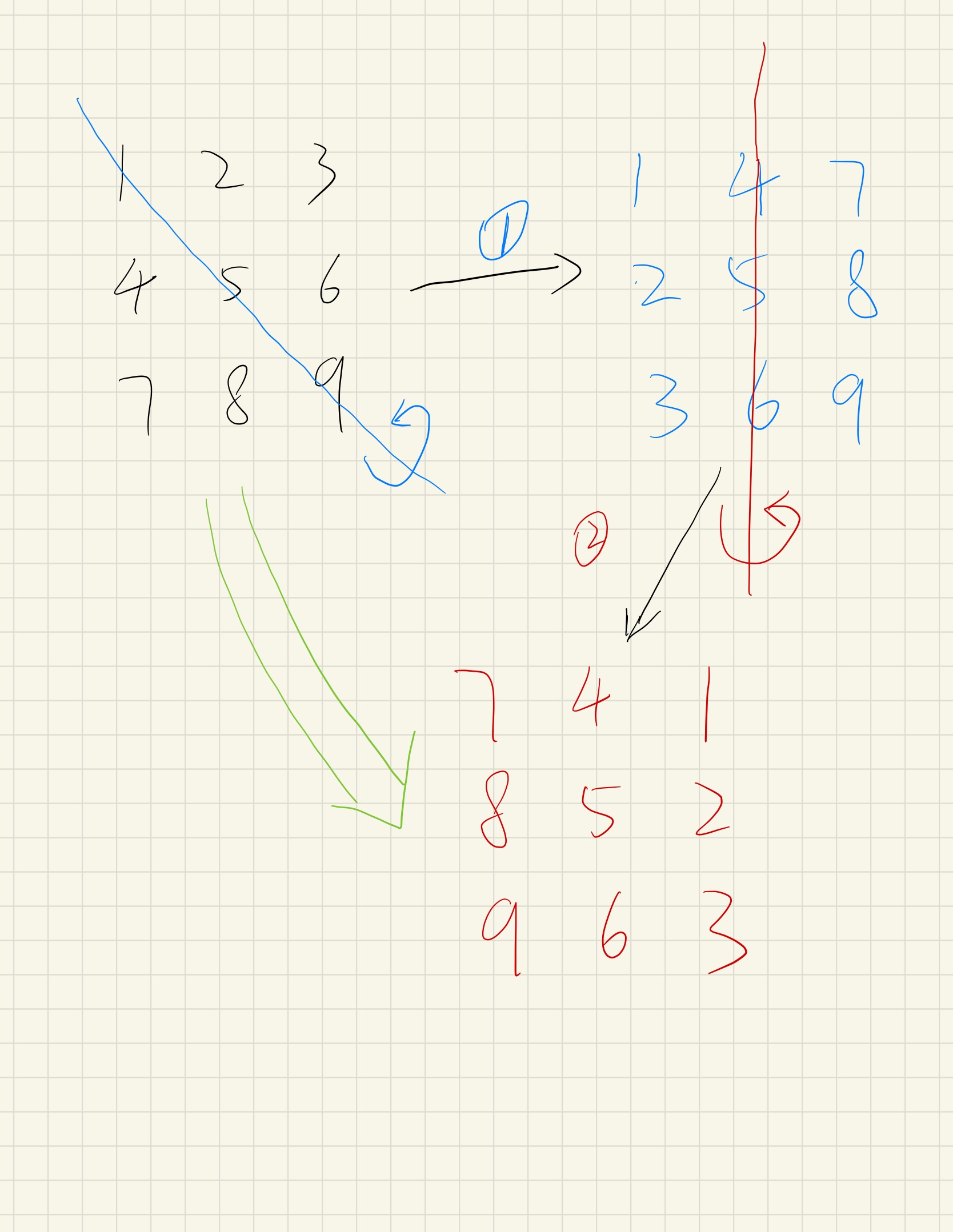
-1000 <= matrix[i][j] <= 1000

# 分析

## 方法一：对角线+倒排

**思路：**

先对矩阵做沿主对角线的翻转（其实就是矩阵的转置），原地完成；接着再对操作后的矩阵进行根据列序号倒排。



**代码：**

class Solution {

public:

    //先按照主对角线翻转，再按列号倒排

    void rotate(vector<vector<int>>& matrix) {

        int n = matrix.size();

        for(int i=0 ; i<n ; ++i){

            for(int j = 0 ; j<i ; ++j){

                swap(matrix[i][j], matrix[j][i]);

            }

        }

        for(int i=0 ; i<n ; ++i){

            for(int j=0 ; j<n/2 ; ++j){

                swap(matrix[i][j], matrix[i][n-1-j]);

            }

        }

    }

};

## 方法二：原地交换