[面试题 01.07. 旋转矩阵](https://leetcode-cn.com/problems/rotate-matrix-lcci/)

<https://leetcode-cn.com/problems/rotate-matrix-lcci/>

Mid=n>>1;

这个表示右移一位，即除以二取整。

学到了！

给你一幅由 N × N 矩阵表示的图像，其中每个像素的大小为 4 字节。请你设计一种算法，将图像旋转 90 度。

不占用额外内存空间能否做到？

示例 1:

给定 matrix =

[

[1,2,3],

[4,5,6],

[7,8,9]

],

原地旋转输入矩阵，使其变为:

[

[7,4,1],

[8,5,2],

[9,6,3]

]

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/rotate-matrix-lcci

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

我的代码：

**class** Solution {  
  
 *//First Try: pure number* **public void** rotatePureNumber(**int**[][] matrix) {  
 **if** (matrix.**length** == 0) {  
 **return**;  
 } **else if** (matrix.**length** == 2) {  
 **int** temp;  
 **int** i = matrix.**length** - 2;  
 temp = matrix[0][0];  
 matrix[0][0] = matrix[1][0];  
 matrix[1][0] = matrix[1][1];  
 matrix[1][1] = matrix[0][1];  
 matrix[0][1] = temp;  
 **return**;  
 } **else if** (matrix.**length** == 3) {  
 **int** temp;  
 temp = matrix[0][0];  
 matrix[0][0] = matrix[2][0];  
 matrix[2][0] = matrix[2][2];  
 matrix[2][2] = matrix[0][2];  
 matrix[0][2] = temp;  
 temp = matrix[0][1];  
 matrix[0][1] = matrix[1][0];  
 matrix[1][0] = matrix[2][1];  
 matrix[2][1] = matrix[1][2];  
 matrix[1][2] = temp;  
 **return**;  
 } **else if** (matrix.**length** == 4) {  
 **int** temp;  
 temp = matrix[0][0];  
 matrix[0][0] = matrix[3][0];  
 matrix[3][0] = matrix[3][3];  
 matrix[3][3] = matrix[0][3];  
 matrix[0][3] = temp;  
 temp = matrix[0][1];  
 matrix[0][1] = matrix[2][0];  
 matrix[2][0] = matrix[3][2];  
 matrix[3][2] = matrix[1][3];  
 matrix[1][3] = temp;  
 temp = matrix[0][2];  
 matrix[0][2] = matrix[1][0];  
 matrix[1][0] = matrix[3][1];  
 matrix[3][1] = matrix[2][3];  
 matrix[2][3] = temp;  
 *//in* temp = matrix[1][1];  
 matrix[1][1] = matrix[2][1];  
 matrix[2][1] = matrix[2][2];  
 matrix[2][2] = matrix[1][2];  
 matrix[1][2] = temp;  
 **return**;  
 } **else** {  
 **return**;  
 }  
 }  
  
 **public void** ratateLeetcode(**int**[][] matrix) {  
 **int** n = matrix.**length**;  
 *// 先以对角线（左上-右下）为轴进行翻转* **for** (**int** i = 0; i < n - 1; i++) {  
 **for** (**int** j = i + 1; j < n; j++) {  
 **int** tmp = matrix[i][j];  
 matrix[i][j] = matrix[j][i];  
 matrix[j][i] = tmp;  
 }  
 }  
 *// 再对每一行以中点进行翻转* **int** mid = n >> 1;  
 **for** (**int** i = 0; i < n; i++) {  
 **for** (**int** j = 0; j < mid; j++) {  
 **int** tmp = matrix[i][j];  
 matrix[i][j] = matrix[i][n - 1 - j];  
 matrix[i][n - 1 - j] = tmp;  
 }  
 }  
 }  
  
 **public void** rotateMyLeetCode(**int**[][] matrix) {  
 **int** n = matrix.**length**;  
 **int** midN = n >> 1;  
 **int** temp;  
  
 *//对角线置换* **for** (**int** i = 0; i < n; i++) {  
 **for** (**int** j = i; j < n; j++) {  
 temp = matrix[i][j];  
 matrix[i][j] = matrix[j][i];  
 matrix[j][i] = temp;  
 }  
 }  
  
 *//左右置换* **for** (**int** i = 0; i < n; i++) {  
 **for** (**int** j = 0; j < midN; j++) {  
 temp = matrix[i][j];  
 matrix[i][j] = matrix[i][n - 1 - j];  
 matrix[i][n - 1 - j] = temp;  
 }  
 }  
 }  
  
 **public void** rotate(**int**[][] matrix) {  
 rotateMyLeetCode(matrix);  
 }  
  
  
 **public static void** main(String[] args) {  
 Solution s = **new** Solution();  
 **int** N = 4;  
 **int**[][][] mList = **new int**[N][][];  
  
 mList[0] = **new int**[][]{  
 {1, 2},  
 {3, 4}};  
 mList[1] = **new int**[][]{  
 {1, 2, 3},  
 {4, 5, 6},  
 {7, 8, 9}};  
 mList[2] = **new int**[][]{  
 {1, 2, 3, 4},  
 {5, 6, 7, 8},  
 {9, 10, 11, 12},  
 {13, 14, 15, 16}};  
 mList[3] = **new int**[][]{  
 {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}};  
  
 **for** (**int** i = 0; i < N; i++) {  
 s.rotate(mList[i]);  
 **for** (**int**[] list : mList[i]) {  
 **for** (**int** num : list) {  
 System.***out***.print(num + **" "**);  
 }  
 System.***out***.println();  
 }  
 System.***out***.println(**"-----"**);  
 }  
  
 }  
}