

一、题目说明

题目是48. Rotate Image，简而言之就是矩阵顺时针旋转90度。不允许使用额外的矩阵。

经过观察（写一个矩阵，多看几遍就知道了），旋转90度后：

第1行变为len-1列（最后一列），第i行变为 len-i-1列最终达到(i,j)-->(j,len-i-1)

知道规律后，做法就比较简单了，可以先上下行交换，再对角交换：

(i,j)-->(len-i-1,j)-->(j,len-i-1)

也可以先对角交换，再左右列交换：

(i,j)-->(j,i)-->(j,len-i-1)

二、我的做法

先上下行交换，再对角交换，(i,j)-->(len-i-1,j)-->(j,len-i-1)，代码如下：

```
class Solution{
public:
    void rotate(vector<vector<int>>& matrix){
        //直接观察，第1行变为n-1列，第i行变为 len-i-1列
        //最终达到(i,j)-->(j,len-i-1)
        //可以： 先上下交换再对角交换 (i,j)-->(len-i-1,j)-->(j,len-i-1)

        int temp,len = matrix.size();

        for(int i = 0;i < len / 2;++i)
        {
            swap(matrix[i],matrix[len - 1 - i]);
        }

        for(int i = 0;i < len;++i)
        {
            for(int j = 0;j < i;++j)
            {
                swap(matrix[i][j],matrix[j][i]);
            }
        }
    }
};
```

代码性能如下：

```
Runtime: 4 ms, faster than 83.46% of C++ online submissions for Rotate Image.
Memory Usage: 9.1 MB, less than 60.98% of C++ online submissions for Rotate Image.
```

三、优化措施

先对角交换，再左右列交换，(i,j)-->(j,i)-->(j,len-i-1)，代码如下：

```

class Solution{
public:
    void rotate(vector<vector<int>>& matrix){
        //直接观察，第1行变为n-1列，第i行变为 len-i-1列
        //最终达到(i,j)-->(j,len-i-1)
        //可以： 先按斜对角线x=y交换，再按垂直中线左右交换
        // (i,j)-->(j,i)-->(j,len-i-1)

        int tmp;
        int i,j,len=matrix.size();
        for(i=0;i<len;i++){
            for(j=i;j<len;j++){
                tmp = matrix[i][j];
                matrix[i][j] = matrix[j][i];
                matrix[j][i] = tmp;
            }
        }

        for(i=0; i<len; i++)
            for(j=0; j<=(len-1)/2; j++){
                tmp = matrix[i][j];
                matrix[i][j] = matrix[i][len-1-j];
                matrix[i][len-1-j] = tmp;
            }
    }
};

```

代码性能:

Runtime: 8 ms, faster than 17.16% of C++ online submissions for Rotate Image.
 Memory Usage: 8.9 MB, less than 100.00% of C++ online submissions for Rotate Image.