```
/*
给定一个 n × n 的二维矩阵表示一个图像。
将图像顺时针旋转 90 度。
说明：
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:
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]
]
示例 2:
给定 matrix =
[
  [ 5, 1, 9,11],
  [ 2, 4, 8,10],
  [13, 3, 6, 7],
  [15,14,12,16]
],
原地旋转输入矩阵，使其变为:
[
  [15,13, 2, 5],
  [14, 3, 4, 1],
  [12, 6, 8, 9],
  [16, 7,10,11]
]
来源：力扣（LeetCode）
链接：https://leetcode-cn.com/problems/rotate-image
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。
*/
```

分析:

- 原地顺时针旋转90步骤:
  - 1.行变换:以行中心对折
  - 2.转置

方法一:C++_原地法

```
class Solution
{
    public:
```

```cpp
        void rotate(vector<vector<int>>& matrix)
        {
            int rows = matrix.size();
            int cols = rows;
            int i    = 0;
            int j    = 0;

            /*1.按行中心对折*/
            for(i = 0 ; i < rows/2 ; i++)
            {
                swap(matrix[i],matrix[rows-1-i]);
            }

            /*2.转置*/
            for(i = 0 ; i < rows ; i++)
            {
                for(j=i+1;j<cols;j++)
                {
                    swap(matrix[i][j],matrix[j][i]);
                }
            }
        }
};
/*
执行结果:
通过
显示详情
执行用时 :4 ms, 在所有 cpp 提交中击败了96.63%的用户
内存消耗 :9 MB, 在所有 cpp 提交中击败了83.04%的用户
*/
```

AlimyBreak
2019.10.16