1. Rotate Image

You are given an *n* x *n* 2D matrix representing an image.

Rotate the image by 90 degrees (clockwise).

**Note:**

You have to rotate the image [**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm), which means you have to modify the input 2D matrix directly. **DO NOT** allocate another 2D matrix and do the rotation.

**Example 1:**

Given input matrix =   
[  
 [1,2,3],  
 [4,5,6],  
 [7,8,9]  
],  
  
rotate the input matrix in-place such that it becomes:  
[  
 [7,4,1],  
 [8,5,2],  
 [9,6,3]  
]

**Example 2:**

Given input matrix =  
[  
 [ 5, 1, 9,11],  
 [ 2, 4, 8,10],  
 [13, 3, 6, 7],  
 [15,14,12,16]  
],   
  
rotate the input matrix in-place such that it becomes:  
[  
 [15,13, 2, 5],  
 [14, 3, 4, 1],  
 [12, 6, 8, 9],  
 [16, 7,10,11]  
]

**解**

先转置，再翻转

class Solution {  
public:  
 void rotate(vector<vector<int>>& matrix) {  
 for(int i = 0; i < matrix.size(); ++i){  
 for(int j = i + 1; j < matrix.size(); ++j){  
 swap(matrix[i][j], matrix[j][i]);  
 }  
 }  
 for(int i = 0; i < matrix.size(); ++i){  
 reverse(matrix[i].begin(), matrix[i].end());  
 }  
 }  
};