48. Rotate Image

Medium

1266124FavoriteShare

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]  
]

class Solution {

public:

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

int n=matrix[0].size();

//diagonal flip

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

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

int temp=matrix[i][j];

matrix[i][j]=matrix[j][i];

matrix[j][i]=temp;

}

}

//reverse each row

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

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

int temp=matrix[i][j];

matrix[i][j]=matrix[i][n-1-j];

matrix[i][n-1-j]=temp;

}

}

}

};

Success

[Details](https://leetcode.com/submissions/detail/207706915/)

Runtime: 8 ms, faster than 100.00% of C++ online submissions for Rotate Image.

Memory Usage: 9.3 MB, less than 0.67% of C++ online submissions forRotate Image.