30 Days Leetcode challenge - Day 12

import java.util.ArrayList;

import java.util.List;

public class leetday12

{

class Solution1 {

public void rotate(int[][] arr)

{

int n = arr.length;

int m = arr[0].length;

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

{

for(int j=i+1;j<m;j++)

{

swap(arr,i,j);

}

}

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

{

reverse(arr[i]);

}

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

{

for(int j=0;j<m;j++)

{

System.out.print(arr[i][j]+" ");

}

System.out.println();

}

}

public static void swap ( int arr [][] , int i , int j)

{

int temp = arr [i][j];

arr[i][j] = arr[j][i];

arr[j][i] = temp;

}

public static void reverse ( int arr [])

{

int left = 0;

int right = arr.length -1 ;

while(left<right)

{

int temp = arr[left] ;

arr[left] = arr[right];

arr[right] = temp;

left++;

right--;

}

}

}

class Solution2 {

public List<Integer> spiralOrder(int[][] arr) {

List<Integer> result = new ArrayList<>();

int n = arr.length;

int m = arr[0].length;

int top = 0, left = 0;

int right = m - 1, bottom = n - 1;

while (top <= bottom && left <= right) {

for (int i = left; i <= right; i++) {

result.add(arr[top][i]);

}

top++;

for (int i = top; i <= bottom; i++) {

result.add(arr[i][right]);

}

right--;

if (top <= bottom) {

for (int i = right; i >= left; i--) {

result.add(arr[bottom][i]);

}

bottom--;

}

if (left <= right) {

for (int i = bottom; i >= top; i--) {

result.add(arr[i][left]);

}

left++;

}

}

return result;

}

}

}