Rotate Matrix

Approach

(Introduction to Problem Solving - II)

Q5. Rotate Matrix </> Solved

Using hints except Complete Solution is Penalty free now — Use Hint

Problem Description

You are given a n x n 2D matrix **A** representing an image.
Rotate the image by 90 degrees (clockwise).
You need to do this in place.
Note: If you end up using an additional array, you will only receive partial score.

Problem Constraints

1 <= n <= 1000

Input Format

First argument is a 2D matrix A of integers

Output Format

Return the 2D rotated matrix.

Example Input

Input 1:

```
[
   [1, 2],
   [3, 4]
]
```

Input 2:

```
[
   [1, 2, 3],
   [4, 5, 6],
   [7, 8, 9]
]
```

Example Output

Output 1:

```
[
   [3, 1],
   [4, 2]
]
```

Output 2:

```
[
   [7, 4, 1],
   [8, 5, 2],
   [9, 6, 3]
]
```

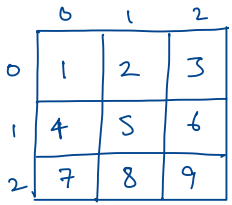
Example Explanation

Explanation 1:

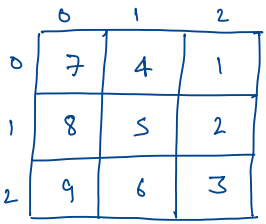
```
After rotating the matrix by 90 degree:
1 goes to 2, 2 goes to 4
4 goes to 3, 3 goes to 1
```

Explanation 2:

```
After rotating the matrix by 90 degree:
1 goes to 3, 3 goes to 9
2 goes to 6, 6 goes to 8
9 goes to 7, 7 goes to 1
8 goes to 4, 4 goes to 2
```



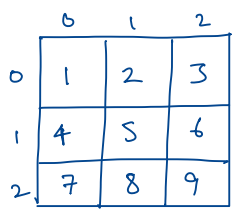
90°



→ It's mentioned that we have to make changes in the original matrix itself

→ It's also given that it is a square matrix

→ We can create the transpose of a square matrix in the original itself in $O(N^2)$ time



Transpose

1	4	7
2	5	8
3	6	9

Observe the current state of this matrix

→ If we reverse all the row elements we will have gotten the matrix that we originally wanted!

90°

reverse row-elements

7	4	1
8	5	2
9	6	3

N = 3

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

1	4	7
2	5	8
3	6	9

N = 4

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

Code Skeleton

```
// Transpose square matrix

for (i = 0; i < N-2; i++) {

    for (j = i+2; j < N; j++) {

        int temp = A[i][j];
        A[i][j] = A[j][i];
        A[j][i] = temp;

    }

}
```

```
// Reverse elements of each row

void reverseRow(int[] arr) {
    int si = 0, ei = arr.length-2;
    while(si < ei) {
        int temp = arr[si];
        arr[si] = arr[ei];
        arr[ei] = temp;
    }
}

void reverseMatrix(int[][] A) {
    for(i = 0; i < A.length; i++) {
        reverseRow(A[i]);
    }
}
```