WizData Interview Tasks

Test for .NET Developer Applicants

Roman Nofenko

2/24/2017

This assessment will be used to evaluate skills and knowledge of all applicants and serve as one factor in considering employment.

Environment and Implementation

This task can be completed in C# .NET 4.6 or in JavaScript.

- Task should be done in non-static classes.
- Entrance class should inherit interface IMatrixRotator
- Result should be in one file with name InterviewMatrixRotator.cs or InterviewMatrixRotator.js
- Implementation should be sent to rnofenko@wdsystems.com
- Time for implementation: up to 5 days. But it’s better to do it as earlier as possible.

Material for implementation

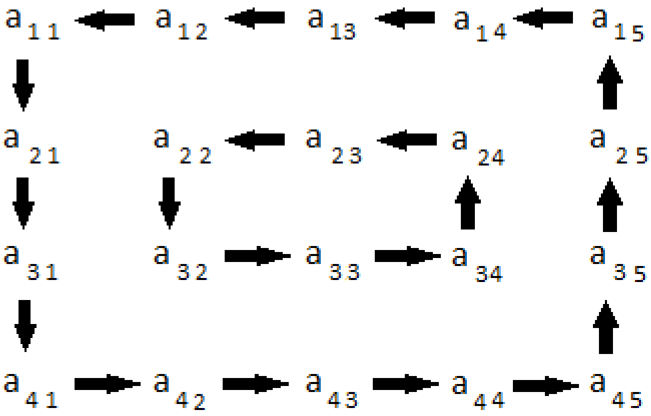
```
public interface IMatrixRotator
    {
        /// <summary>
        ///
        /// </summary>
        /// <param name="martix">Initial matrix</param>
        /// <param name="r">Number of rotations</param>
        /// <returns>Rotated matrix</returns>
        int[][] Rotate(int[][] martix, int r);
    }
```

Criteria for implementation

1. Solves an example in this document below
2. Passes tests with different matrix variations
3. Passes performance tests with the fastest time possible
4. Good, readable code

Task description

You are given a 2D matrix, *a*, of dimension *MxN* and a positive integer, *R*. You have to rotate the matrix *R* times and return the resultant matrix. Rotations should be in counter-clockwise direction. Rotation of a *4x5* matrix is represented by the following figure.



Matrix Rotation

It is guaranteed that the minimum of *M* and *N* will be even.

Constraints

```
2 <= M, N <= 300

1 <= R <= 109

min(M, N) % 2 == 0

1 <= aij <= 108, where i ∈ [1..M] & j ∈ [1..N]
```

Sample Input #00

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

Sample Output #00

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

Sample Input #01

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

Sample Output #01

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

Sample Input #02

```
7
1 2 3 4
7 8 9 10
13 14 15 16
19 20 21 22
25 26 27 28
```

Sample Output #02

```
28 27 26 25
22 9 15 19
16 8 21 13
10 14 20 7
4 3 2 1
```

Sample Input #03

```
3
1 1
1 1
```

Sample Output #03

```
1 1
1 1
```

Explanation

Sample Case #00

Here is an illustration of what happens when the matrix is rotated once.

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

Sample Case #01

Here is what happens to the matrix after two rotations.

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

Sample Case #02

Following are the intermediate states.

```
1  2  3  4      2  3  4 10    3  4 10 16    4 10 16 22
7  8  9 10      1  9 15 16    2 15 21 22    3 21 20 28
13 14 15 16 ->  7  8 21 22 -> 1  9 20 28 -> 2 15 14 27 ->
19 20 21 22    13 14 20 28    7  8 14 27    1  9  8 26
25 26 27 28    19 25 26 27    13 19 25 26   7 13 19 25



10 16 22 28    16 22 28 27    22 28 27 26    28 27 26 25
 4 20 14 27    10 14  8 26    16  8  9 25    22  9 15 19
 3 21  8 26 ->  4 20  9 25 -> 10 14 15 19 -> 16  8 21 13
 2 15  9 25     3 21 15 19     4 20 21 13    10 14 20  7
 1  7 13 19     2  1  7 13     3  2  1  7     4  3  2  1
```