BACK

# Contours Shifting

168,654

DESCRIPTION | MY SOLUTIONS | LEADERBOARD | COMMENTS | README

CODEWRITING SCORE: 300/300

Mark got a rectangular array `matrix` for his birthday, and now he's thinking about all the fun things he can do with it. He likes shifting a lot, so he decides to shift all of its *i-contours* in a clockwise direction if `i` is even, and counterclockwise if `i` is odd.

Here is how Mark defines *i-contours*:

- the *0-contour* of a rectangular array as the union of left and right columns as well as top and bottom rows;
- consider the initial matrix without the *0-contour*: its *0-contour* is the *1-contour* of the initial matrix;
- define *2-contour*, *3-contour*, etc. in the same manner by removing *0-contours* from the obtained arrays.

Implement a function that does exactly what Mark wants to do to his matrix.

**Example**

- For

```
matrix = [[ 1,  2,  3,  4],
          [ 5,  6,  7,  8],
          [ 9, 10, 11, 12],
          [13, 14, 15, 16],
          [17, 18, 19, 20]]
```

the output should be

```
contoursShifting(matrix) = [[ 5,  1,  2,  3],
                            [ 9,  7, 11,  4],
                            [13,  6, 15,  8],
                            [17, 10, 14, 12],
                            [18, 19, 20, 16]]
```

- For `matrix = [[238, 239, 240, 241, 242, 243, 244, 245]]`,
the output should be
`contoursShifting(matrix) = [[245, 238, 239, 240, 241, 242, 243, 244]]`.

Note, that if a *contour* is represented by a `1 × n` array, its center is considered to be *below* it.

- For

```
matrix = [[238],
          [239],
          [240],
          [241],
          [242],
          [243],
          [244],
          [245]]
```

the output should be

```
contoursShifting(matrix) = [[245],
                            [238],
                            [239],
                            [240],
                            [241],
                            [242],
                            [243],
                            [244]]
```

If a *contour* is represented by an `n × 1` array, its center is considered to be *to the left* of it.

**Input/Output**