Problem 1914: Cyclically Rotating a Grid

Problem Information

Difficulty: Medium
Acceptance Rate: 51.31%
Paid Only: No
Tags: Array, Matrix, Simulation

Problem Description

You are given an `m x n` integer matrix `grid`■■■, where `m` and `n` are both **even** integers, and an integer `k`.

The matrix is composed of several layers, which is shown in the below image, where each color is its own layer:

A cyclic rotation of the matrix is done by cyclically rotating **each layer** in the matrix. To cyclically rotate a layer once, each element in the layer will take the place of the adjacent element in the **counter-clockwise** direction. An example rotation is shown below:

Return _the matrix after applying_`k` _cyclic rotations to it_.

Example 1:

Input: grid = [[40,10],[30,20]], k = 1 **Output:** [[10,20],[40,30]] **Explanation:** The figures above represent the grid at every state.

Example 2:

(https://assets.leetcode.com/uploads/2021/06/10/ringofgrid5.png)**(https://assets.leetcode.com/uploads/2021/06/10/ringofgrid6.png)****(https://assets.leetcode.com/uploads/2021/06/10/ringofgrid7.png)**

Input: grid = [[1,2,3,4],[5,6,7,8],[9,10,11,12],[13,14,15,16]], k = 2 **Output:** [[3,4,8,12],[2,11,10,16],[1,7,6,15],[5,9,13,14]] **Explanation:** The figures above represent the grid at every state.

Constraints:

* `m == grid.length` * `n == grid[i].length` * `2 <= m, n <= 50` * Both `m` and `n` are **even** integers. * `1 <= grid[i][j] <= 5000` * `1 <= k <= 109`

Code Snippets

C++:

```
class Solution {
public:
vector<vector<int>> rotateGrid(vector<vector<int>>& grid, int k) {

}
};
```

Java:

```
class Solution {
public int[][] rotateGrid(int[][] grid, int k) {

}
}
```

Python3:

```
class Solution:
def rotateGrid(self, grid: List[List[int]], k: int) -> List[List[int]]:
```