Problem 2946: Matrix Similarity After Cyclic Shifts

Problem Information

Difficulty: Easy
Acceptance Rate: 58.98%
Paid Only: No
Tags: Array, Math, Matrix, Simulation

Problem Description

You are given an `m x n` integer matrix `mat` and an integer `k`. The matrix rows are 0-indexed.

The following proccess happens `k` times:

* **Even-indexed** rows (0, 2, 4, ...) are cyclically shifted to the left.

![](https://assets.leetcode.com/uploads/2024/05/19/lshift.jpg)

* **Odd-indexed** rows (1, 3, 5, ...) are cyclically shifted to the right.

![](https://assets.leetcode.com/uploads/2024/05/19/rshift-stlone.jpg)

Return `true` if the final modified matrix after `k` steps is identical to the original matrix, and `false` otherwise.

Example 1:

Input: mat = [[1,2,3],[4,5,6],[7,8,9]], k = 4

Output: false

Explanation:

In each step left shift is applied to rows 0 and 2 (even indices), and right shift to row 1 (odd index).

![](https://assets.leetcode.com/uploads/2024/05/19/t1-2.jpg)

Example 2:

Input: mat = [[1,2,1,2],[5,5,5,5],[6,3,6,3]], k = 2

Output: true

Explanation:

![](https://assets.leetcode.com/uploads/2024/05/19/t1-3.jpg)

Example 3:

Input: mat = [[2,2],[2,2]], k = 3

Output: true

Explanation:

As all the values are equal in the matrix, even after performing cyclic shifts the matrix will remain the same.

Constraints:

* `1 <= mat.length <= 25` * `1 <= mat[i].length <= 25` * `1 <= mat[i][j] <= 25` * `1 <= k <= 50`

Code Snippets

C++:

```
class Solution {
public:
bool areSimilar(vector<vector<int>>& mat, int k) {

}
};
```

Java:

```
class Solution {
public boolean areSimilar(int[][] mat, int k) {

}
}
```

Python3:

```
class Solution:
def areSimilar(self, mat: List[List[int]], k: int) -> bool:
```