Problem 1260: Shift 2D Grid

Problem Information

Difficulty: Easy
Acceptance Rate: 67.91%
Paid Only: No
Tags: Array, Matrix, Simulation

Problem Description

Given a 2D `grid` of size `m x n` and an integer `k`. You need to shift the `grid` `k` times.

In one shift operation:

* Element at `grid[i][j]` moves to `grid[i][j + 1]`. * Element at `grid[i][n - 1]` moves to `grid[i + 1][0]`. * Element at `grid[m - 1][n - 1]` moves to `grid[0][0]`.

Return the _2D grid_ after applying shift operation `k` times.

Example 1:

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

Example 2:

Input: grid = [[3,8,1,9],[19,7,2,5],[4,6,11,10],[12,0,21,13]], k = 4 **Output:** [[12,0,21,13],[3,8,1,9],[19,7,2,5],[4,6,11,10]]

Example 3:

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

Constraints:

* `m == grid.length` * `n == grid[i].length` * `1 <= m <= 50` * `1 <= n <= 50` * `-1000 <= grid[i][j] <= 1000` * `0 <= k <= 100`

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public List<List<Integer>> shiftGrid(int[][] grid, int k) {

}
}
```

Python3:

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