Problem 3653: XOR After Range Multiplication Queries I

Problem Information

Difficulty: Medium
Acceptance Rate: 70.40%
Paid Only: No
Tags: Array, Divide and Conquer, Simulation

Problem Description

You are given an integer array `nums` of length `n` and a 2D integer array `queries` of size `q`, where `queries[i] = [li, ri, ki, vi]`.

For each query, you must apply the following operations in order:

* Set `idx = li`. * While `idx <= ri`: * Update: `nums[idx] = (nums[idx] * vi) % (109 + 7)` * Set `idx += ki`.

Return the **bitwise XOR** of all elements in `nums` after processing all queries.

Example 1:

Input: nums = [1,1,1], queries = [[0,2,1,4]]

Output: 4

Explanation:

* A single query `[0, 2, 1, 4]` multiplies every element from index 0 through index 2 by 4. * The array changes from `[1, 1, 1]` to `[4, 4, 4]`. * The XOR of all elements is `4 ^ 4 ^ 4 = 4`.

Example 2:

Input: nums = [2,3,1,5,4], queries = [[1,4,2,3],[0,2,1,2]]

Output: 31

Explanation:

* The first query `[1, 4, 2, 3]` multiplies the elements at indices 1 and 3 by 3, transforming the array to `[2, 9, 1, 15, 4]`. * The second query `[0, 2, 1, 2]` multiplies the elements at indices 0, 1, and 2 by 2, resulting in `[4, 18, 2, 15, 4]`. * Finally, the XOR of all elements is `4 ^ 18 ^ 2 ^ 15 ^ 4 = 31`.■■■■■■■■**■■■■■■■■**

Constraints:

* `1 <= n == nums.length <= 103` * `1 <= nums[i] <= 109` * `1 <= q == queries.length <= 103` * `queries[i] = [li, ri, ki, vi]` * `0 <= li <= ri < n` * `1 <= ki <= n` * `1 <= vi <= 105`

Code Snippets

C++:

```
class Solution {
public:
int xorAfterQueries(vector<int>& nums, vector<vector<int>>& queries) {

}
};
```

Java:

```
class Solution {
public int xorAfterQueries(int[] nums, int[][] queries) {

}
}
```

Python3:

```
class Solution:
def xorAfterQueries(self, nums: List[int], queries: List[List[int]]) -> int:
```