Problem 3489: Zero Array Transformation IV

Problem Information

Difficulty: Medium
Acceptance Rate: 30.59%
Paid Only: No
Tags: Array, Dynamic Programming

Problem Description

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

Each `queries[i]` represents the following action on `nums`:

* Select a subset of indices in the range `[li, ri]` from `nums`. * Decrement the value at each selected index by **exactly** `vali`.

A **Zero Array** is an array with all its elements equal to 0.

Return the **minimum** possible **non-negative** value of `k`, such that after processing the first `k` queries in **sequence** , `nums` becomes a **Zero Array**. If no such `k` exists, return -1.

Example 1:

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

Output: 2

Explanation:

* **For query 0 (l = 0, r = 2, val = 1):** * Decrement the values at indices `[0, 2]` by 1. * The array will become `[1, 0, 1]`. * **For query 1 (l = 0, r = 2, val = 1):** * Decrement the values at indices `[0, 2]` by 1. * The array will become `[0, 0, 0]`, which is a Zero Array. Therefore, the minimum value of `k` is 2.

Example 2:

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

Output: -1

Explanation:

It is impossible to make nums a Zero Array even after all the queries.

Example 3:

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

Output: 4

Explanation:

* **For query 0 (l = 0, r = 1, val = 1):** * Decrement the values at indices `[0, 1]` by `1`. * The array will become `[0, 1, 3, 2, 1]`. * **For query 1 (l = 1, r = 2, val = 1):** * Decrement the values at indices `[1, 2]` by 1. * The array will become `[0, 0, 2, 2, 1]`. * **For query 2 (l = 2, r = 3, val = 2):** * Decrement the values at indices `[2, 3]` by 2. * The array will become `[0, 0, 0, 0, 1]`. * **For query 3 (l = 3, r = 4, val = 1):** * Decrement the value at index 4 by 1. * The array will become `[0, 0, 0, 0, 0]`. Therefore, the minimum value of `k` is 4.

Example 4:

Input: nums = [1,2,3,2,6], queries = [[0,1,1],[0,2,1],[1,4,2],[4,4,4],[3,4,1],[4,4,5]]

Output: 4

Constraints:

* `1 <= nums.length <= 10` * `0 <= nums[i] <= 1000` * `1 <= queries.length <= 1000` * `queries[i] = [li, ri, vali]` * `0 <= li <= ri < nums.length` * `1 <= vali <= 10`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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