Problem 2569: Handling Sum Queries After Update

Problem Information

Difficulty: Hard
Acceptance Rate: 30.06%
Paid Only: No
Tags: Array, Segment Tree

Problem Description

You are given two **0-indexed** arrays `nums1` and `nums2` and a 2D array `queries` of queries. There are three types of queries:

1. For a query of type 1, `queries[i] = [1, l, r]`. Flip the values from `0` to `1` and from `1` to `0` in `nums1` from index `l` to index `r`. Both `l` and `r` are **0-indexed**. 2. For a query of type 2, `queries[i] = [2, p, 0]`. For every index `0 <= i < n`, set `nums2[i] = nums2[i] + nums1[i] * p`. 3. For a query of type 3, `queries[i] = [3, 0, 0]`. Find the sum of the elements in `nums2`.

Return _an array containing all the answers to the third type queries._

Example 1:

Input: nums1 = [1,0,1], nums2 = [0,0,0], queries = [[1,1,1],[2,1,0],[3,0,0]] **Output:** [3] **Explanation:** After the first query nums1 becomes [1,1,1]. After the second query, nums2 becomes [1,1,1], so the answer to the third query is 3. Thus, [3] is returned.

Example 2:

Input: nums1 = [1], nums2 = [5], queries = [[2,0,0],[3,0,0]] **Output:** [5] **Explanation:** After the first query, nums2 remains [5], so the answer to the second query is 5. Thus, [5] is returned.

Constraints:

* `1 <= nums1.length,nums2.length <= 105` * `nums1.length = nums2.length` * `1 <= queries.length <= 105` * `queries[i].length = 3` * `0 <= l <= r <= nums1.length - 1` * `0 <= p <= 106` * `0 <= nums1[i] <= 1` * `0 <= nums2[i] <= 109`

Code Snippets

C++:

```
class Solution {
public:
vector<long long> handleQuery(vector<int>& nums1, vector<int>& nums2,
vector<vector<int>>& queries) {

}
};
```

Java:

```
class Solution {
public long[] handleQuery(int[] nums1, int[] nums2, int[][] queries) {

}
}
```

Python3:

```
class Solution:
def handleQuery(self, nums1: List[int], nums2: List[int], queries:
List[List[int]]) -> List[int]:
```