Problem 3526: Range XOR Queries with Subarray Reversals

Problem Information

Difficulty: Hard
Acceptance Rate: 62.48%
Paid Only: Yes
Tags: Array, Tree, Binary Tree

Problem Description

You are given an integer array `nums` of length `n` and a 2D integer array `queries` of length `q`, where each query is one of the following three types:

1. **Update** : `queries[i] = [1, index, value]` Set `nums[index] = value`.

2. **Range XOR Query** : `queries[i] = [2, left, right]` Compute the bitwise XOR of all elements in the subarray `nums[left...right]`, and record this result.

3. **Reverse Subarray** : `queries[i] = [3, left, right]` Reverse the subarray `nums[left...right]` in place.

Return _an array of the results of all range XOR queries_ in the order they were encountered.

Example 1:

Input: nums = [1,2,3,4,5], queries = [[2,1,3],[1,2,10],[3,0,4],[2,0,4]]

Output: [5,8]

Explanation:

* **Query****1****:** `[2, 1, 3]` - Compute XOR of subarray `[2, 3, 4]` resulting in 5.

* **Query 2:** `[1, 2, 10]` - Update `nums[2]` to 10, updating the array to `[1, 2, 10, 4, 5]`.

* **Query 3:** `[3, 0, 4]` - Reverse the entire array to get `[5, 4, 10, 2, 1]`.

* **Query 4:** `[2, 0, 4]` - Compute XOR of subarray `[5, 4, 10, 2, 1]` resulting in 8.

Example 2:

Input: nums = [7,8,9], queries = [[1,0,3],[2,0,2],[3,1,2]]

Output: [2]

Explanation:

* **Query 1:** `[1, 0, 3]` - Update `nums[0]` to 3, updating the array to `[3, 8, 9]`.

* **Query 2:** `[2, 0, 2]` - Compute XOR of subarray `[3, 8, 9]` resulting in 2.

* **Query 3:** `[3, 1, 2]` - Reverse the subarray `[8, 9]` to get `[9, 8]`.

Constraints:

* `1 <= nums.length <= 105` * `0 <= nums[i] <= 109` * `1 <= queries.length <= 105` * `queries[i].length == 3■` * `queries[i][0] ∈ {1, 2, 3}■` * If `queries[i][0] == 1`:`■` * `0 <= index < nums.length■` * `0 <= value <= 109` * If `queries[i][0] == 2` or `queries[i][0] == 3`:`■` * `0 <= left <= right < nums.length■`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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