Problem 2179: Count Good Triplets in an Array

Problem Information

Difficulty: Hard
Acceptance Rate: 65.81%
Paid Only: No
Tags: Array, Binary Search, Divide and Conquer, Binary Indexed Tree, Segment Tree, Merge Sort, Ordered Set

Problem Description

You are given two **0-indexed** arrays `nums1` and `nums2` of length `n`, both of which are **permutations** of `[0, 1, ..., n - 1]`.

A **good triplet** is a set of `3` **distinct** values which are present in **increasing order** by position both in `nums1` and `nums2`. In other words, if we consider `pos1v` as the index of the value `v` in `nums1` and `pos2v` as the index of the value `v` in `nums2`, then a good triplet will be a set `(x, y, z)` where `0 <= x, y, z <= n - 1`, such that `pos1x < pos1y < pos1z` and `pos2x < pos2y < pos2z`.

Return _the**total number** of good triplets_.

Example 1:

Input: nums1 = [2,0,1,3], nums2 = [0,1,2,3] **Output:** 1 **Explanation:** There are 4 triplets (x,y,z) such that pos1x < pos1y < pos1z. They are (2,0,1), (2,0,3), (2,1,3), and (0,1,3). Out of those triplets, only the triplet (0,1,3) satisfies pos2x < pos2y < pos2z. Hence, there is only 1 good triplet.

Example 2:

Input: nums1 = [4,0,1,3,2], nums2 = [4,1,0,2,3] **Output:** 4 **Explanation:** The 4 good triplets are (4,0,3), (4,0,2), (4,1,3), and (4,1,2).

Constraints:

* `n == nums1.length == nums2.length` * `3 <= n <= 105` * `0 <= nums1[i], nums2[i] <= n - 1` * `nums1` and `nums2` are permutations of `[0, 1, ..., n - 1]`.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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