(/) Explore(/explore/) Problems(/problemset/all/) Interview New (/contest/) Contest Discuss(/discuss/) 🔊 LeetCode is hiring! Apply NOW.🔊 Store (https://leetcode.com/jobs/) ☆ Premium (/subscribe?ref=nb_npl)

5999. Count Good Triplets in an Array

My Submissions (/contest/biweekly-contest-72/problems/count-good-triplets-in-an-array/submissions/) Back to Contest (/contest/biweekly-contest-72/)

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Hard

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 $pos1_v$ as the index of the value `v` in `nums1` and $pos2_v$ 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 $pos1_x < pos1_y < pos1_z$ and $pos2_x < pos2_y < pos2_z$.

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`
- $3 <= n <= 10^5$
- `0 <= nums1[i], nums2[i] <= n - 1`
- `nums1` and `nums2` are permutations of `[0, 1, ..., n - 1]`.

JavaScript

```javascript
function Fenwick(n) {
    let tree = Array(n).fill(0);
    return { query, update }
    function query(i) {
        let sum = 0;
        i++;
        while (i > 0) {
            sum += tree[i];
            i -= i & -i;
        }
        return sum;
    }
    function update(i, v) {
        i++;
        while (i < n) {
            tree[i] += v;
            i += i & -i;
        }
    }
}

const goodTriplets = (a, b) => {
    let n = a.length, m = new Map(), res = 0;
    for (let i = 0; i < n; i++) m.set(b[i], i);
    let fen = new Fenwick(n + 3);
    for (let i = 0; i < n; i++) {
       let pos = m.get(a[i]);
```

```
        let l = fen.query(pos), r = (n - 1 - pos) - (fen.query(n - 1) - fen.query(pos));
        res += l * r;
        fen.update(pos, 1);
    }
    return res;
};
```

☐ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/644649365/) More Details (/submissions/detail/644649365/)

Share your acceptance!

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)