

6894. Sum of Imbalance Numbers of All Subarrays

My Submissions (/contest/weekly-contest-352/problems/sum-of-imbalance-numbers-of-all-subarrays/submissions/)

Back to Contest (/contest/weekly-contest-352/)

The **imbalance number** of a **0-indexed** integer array `arr` of length `n` is defined as the number of indices `i` in `sarr = sorted(arr)` such that:

- `0 <= i < n - 1`, and
- `sarr[i+1] - sarr[i] > 1`

Here, `sorted(arr)` is the function that returns the sorted version of `arr`.

Given a **0-indexed** integer array `nums`, return *the sum of imbalance numbers of all its subarrays*.

A **subarray** is a contiguous **non-empty** sequence of elements within an array.

| | |
|--------------------|------|
| User Accepted: | 746 |
| User Tried: | 1781 |
| Total Accepted: | 840 |
| Total Submissions: | 3463 |
| Difficulty: | Hard |

Example 1:

Input: `nums = [2,3,1,4]`
Output: `3`
Explanation: There are 3 subarrays with non-zero imbalance numbers:
- Subarray `[3, 1]` with an imbalance number of 1.
- Subarray `[3, 1, 4]` with an imbalance number of 1.
- Subarray `[1, 4]` with an imbalance number of 1.
The imbalance number of all other subarrays is 0. Hence, the sum of imbalance numbers of all the subarrays of `nums` is 3.

Example 2:

Input: `nums = [1,3,3,3,5]`
Output: `8`
Explanation: There are 7 subarrays with non-zero imbalance numbers:
- Subarray `[1, 3]` with an imbalance number of 1.
- Subarray `[1, 3, 3]` with an imbalance number of 1.
- Subarray `[1, 3, 3, 3]` with an imbalance number of 1.
- Subarray `[1, 3, 3, 3, 5]` with an imbalance number of 2.
- Subarray `[3, 3, 3, 5]` with an imbalance number of 1.
- Subarray `[3, 3, 5]` with an imbalance number of 1.
- Subarray `[3, 5]` with an imbalance number of 1.
The imbalance number of all other subarrays is 0. Hence, the sum of imbalance numbers of all the subarrays of `nums` is 8.

Constraints:

- `1 <= nums.length <= 1000`
- `1 <= nums[i] <= nums.length`

Discuss (<https://leetcode.com/problems/sum-of-imbalance-numbers-of-all-subarrays/discuss>)

Java


📄

↺

⚙️


```
1 class Solution {
2     public int sumImbalanceNumbers(int[] a) {
3         int n = a.length;
4         int res = 0;
5         for (int i = 0; i < n; i++) {
6             TreeMap<Integer, Integer> m = new TreeMap<>();
7             int cnt = -1;
8             for (int j = i; j < n; j++) {
9                 int v = a[j];
10                if (!m.containsKey(v)) {
11                    m.merge(v, 1, Integer::sum);
12                    cnt++;
13                    if (m.containsKey(v - 1)) cnt--;
14                    if (m.containsKey(v + 1)) cnt--;
```

```
15         }
16         res += cnt;
17     }
18 }
19 return res;
20 }
21 }
```

☐ Custom Testcase[Use Example Testcases](#)[Run](#)[Submit](#)**Submission Result: Accepted** (</submissions/detail/984385405/>) [More Details >](/submissions/detail/984385405/)

Share your acceptance!

Copyright © 2023 LeetCode

[Help Center \(/support\)](/support/) | [Jobs \(/jobs\)](/jobs/) | [Bug Bounty \(/bugbounty\)](/bugbounty/) | [Online Interview \(/interview/\)](/interview/) | [Students \(/student\)](/student/) | [Terms \(/terms\)](/terms/) | [Privacy Policy \(/privacy\)](/privacy/) [United States \(/region\)](/region/)