

100137. Count Subarrays Where Max Element Appears at Least K Times

My Submissions (/contest/weekly-contest-375/problems/count-subarrays-where-max-element-appears-at-least-k-times/submissions/)

Back to Contest (/contest/weekly-contest-375/)

You are given an integer array `nums` and a **positive** integer `k`.

Return the number of subarrays where the **maximum** element of `nums` appears **at least** `k` times in that subarray.

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

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

Example 1:

**Input:** `nums = [1,3,2,3,3], k = 2`  
**Output:** 6  
**Explanation:** The subarrays that contain the element 3 at least 2 times are: `[1,3,2,3]`, `[3,2,3]`, `[2,3,3]`, `[3,3]`, `[3,3]`, and `[3,3]`.

Example 2:

**Input:** `nums = [1,4,2,1], k = 3`  
**Output:** 0  
**Explanation:** No subarray contains the element 4 at least 3 times.

Constraints:

- 1 <= `nums.length` <= 10<sup>5</sup>
- 1 <= `nums[i]` <= 10<sup>6</sup>
- 1 <= `k` <= 10<sup>5</sup>

JavaScript

📄 ↺ ⚙

```
1 function Bisect() {
2   return { insert_right, insert_left, bisect_left, bisect_right }
3 }
4 function insert_right(a, x, lo = 0, hi = null) {
5   lo = bisect_right(a, x, lo, hi);
6   a.splice(lo, 0, x);
7 }
8 function bisect_right(a, x, lo = 0, hi = null) { // > upper_bound
9   if (lo < 0) throw new Error('lo must be non-negative');
10  if (hi == null) hi = a.length;
11  while (lo < hi) {
12    let mid = parseInt((lo + hi) / 2);
13    a[mid] > x ? hi = mid : lo = mid + 1;
14  }
15  return lo;
16 }
17 function insert_left(a, x, lo = 0, hi = null) {
18   lo = bisect_left(a, x, lo, hi);
19   a.splice(lo, 0, x);
20 }
21 function bisect_left(a, x, lo = 0, hi = null) { // >= lower_bound
22   if (lo < 0) throw new Error('lo must be non-negative');
23   if (hi == null) hi = a.length;
24   while (lo < hi) {
25     let mid = parseInt((lo + hi) / 2);
26     a[mid] < x ? lo = mid + 1 : hi = mid;
27   }
28   return lo;
29 }
```

```
29 }
30
31 const countSubarrays = (a, k) => {
32   let max = Math.max(...a), ia = [], bi = new Bisect(), res = 0;
33   a.map((x, i) => {
34     if (x == max) ia.push(i);
35   })
36   a.map((x, i) => {
37     let idx = bi.bisect_left(ia, i), atLeast = idx + k - 1, startIdx = ia[atLeast];
38     if (startIdx != undefined) {
39       cnt = a.length - startIdx;
40       res += cnt;
41     }
42   })
43   return res;
44 };
```

☐ Custom Testcase[Use Example Testcases](#)[Run](#)[Submit](#)**Submission Result:** Accepted (/submissions/detail/1116180943/) ?[More Details > \(/submissions/detail/1116180943/\)](#)

Share your acceptance!

Copyright © 2023 LeetCode

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