# 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],
```

**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 <= 10^5`
- `1 <= nums[i] <= 10^6`
- `1 <= k <= 10^5`

JavaScript

```
function Bisect() {
    return { insort_right, insort_left, bisect_left, bisect_right }
    function insort_right(a, x, lo = 0, hi = null) {
        lo = bisect_right(a, x, lo, hi);
        a.splice(lo, 0, x);
    }
    function bisect_right(a, x, lo = 0, hi = null) { // > upper_bound
        if (lo < 0) throw new Error('lo must be non-negative');
        if (hi == null) hi = a.length;
        while (lo < hi) {
            let mid = parseInt((lo + hi) / 2);
            a[mid] > x ? hi = mid : lo = mid + 1;
        }
        return lo;
    }
    function insort_left(a, x, lo = 0, hi = null) {
        lo = bisect_left(a, x, lo, hi);
        a.splice(lo, 0, x);
    }
    function bisect_left(a, x, lo = 0, hi = null) { // >= lower_bound
        if (lo < 0) throw new Error('lo must be non-negative');
        if (hi == null) hi = a.length;
        while (lo < hi) {
            let mid = parseInt((lo + hi) / 2);
            a[mid] < x ? lo = mid + 1 : hi = mid;
        }
        return lo;
    }
```

```
}

const countSubarrays = (a, k) => {
    let max = Math.max(...a), ia = [], bi = new Bisect(), res = 0;
    a.map((x, i) => {
        if (x == max) ia.push(i);
    })
    a.map((x, i) => {
        let idx = bi.bisect_left(ia, i), atLeast = idx + k - 1, startIdx = ia[atLeast];
        if (startIdx != undefined) {
            cnt = a.length - startIdx;
            res += cnt;
        }
    })
    return res;
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/1116180943/) | More Details > (/submissions/detail/1116180943/)

Share your acceptance!