6230. Maximum Sum of Distinct Subarrays With Length K

My Submissions (/contest/weekly-contest-318/problems/maximum-sum-of-distinct-subarrays-with-length-k/submissions/)

Back to Contest (/contest/weekly-contest-318/)

You are given an integer array `nums` and an integer `k`. Find the maximum subarray sum of all the subarrays of `nums` that meet the following conditions:

- The length of the subarray is `k`, and
- All the elements of the subarray are **distinct**.

Return *the maximum subarray sum of all the subarrays that meet the conditions.* If no subarray meets the conditions, return `0`.

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

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

Example 1:

```
Input: nums = [1,5,4,2,9,9,9], k = 3
Output: 15
Explanation: The subarrays of nums with length 3 are:
- [1,5,4] which meets the requirements and has a sum of 10.
- [5,4,2] which meets the requirements and has a sum of 11.
- [4,2,9] which meets the requirements and has a sum of 15.
- [2,9,9] which does not meet the requirements because the element 9 is repeated.
- [9,9,9] which does not meet the requirements because the element 9 is repeated.
We return 15 because it is the maximum subarray sum of all the subarrays that meet the conditions
```

Example 2:

```
Input: nums = [4,4,4], k = 3
Output: 0
Explanation: The subarrays of nums with length 3 are:
- [4,4,4] which does not meet the requirements because the element 4 is repeated.
We return 0 because no subarrays meet the conditions.
```

Constraints:

- $1 <= k <= nums.length <= 10^5$
- $1 <= nums[i] <= 10^5$

JavaScript

```
const addOneOrManyMap = (m, x, cnt = 1) => m.set(x, m.get(x) + cnt || cnt);
const removeOneOrManyMap = (m, x, cnt = 1) => { let occ = m.get(x); occ > cnt ? m.set(x, occ - cnt) : m.delete(x); };

const maximumSubarraySum = (a, k) => {
    let n = a.length, res = 0, m = new Map(), sum = 0;
    for (let i = 0; i < n; i++) {
        sum += a[i];
        addOneOrManyMap(m, a[i])
        if (i >= k) {
           sum -= a[i-k];
           removeOneOrManyMap(m, a[i-k]);
        }
        if (m.size == k) res = Math.max(res, sum);
    }
    return res;
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/837805117/) | More Details > (/submissions/detail/837805117/)

Share your acceptance!

3