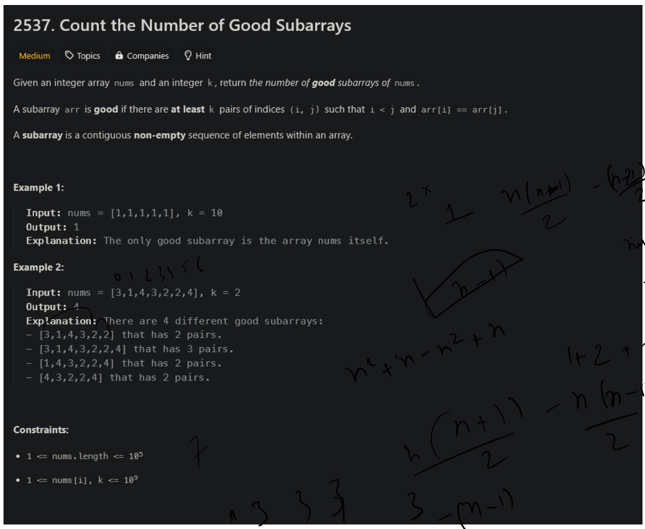
Wednesday, April 16, 2025 7:56 PM

Count the No. of Good Subarrays:-



2537. Count the Number of Good Subarrays

Medium · Topics · Companies · Hint

Given an integer array `nums` and an integer `k`, return *the number of **good** subarrays of* `nums`.

A subarray `arr` is **good** if there are **at least** `k` pairs of indices `(i, j)` such that `i < j` and `arr[i] == arr[j]`.

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

Example 1:

```
Input: nums = [1,1,1,1,1], k = 10
Output: 1
Explanation: The only good subarray is the array nums itself.
```

Example 2:

```
Input: nums = [3,1,4,3,2,2,4], k = 2
Output: 4
Explanation: There are 4 different good subarrays:
- [3,1,4,3,2,2] that has 2 pairs.
- [3,1,4,3,2,2,4] that has 3 pairs.
- [1,4,3,2,2,4] that has 2 pairs.
- [4,3,2,2,4] that has 2 pairs.
```

Constraints:

- `1 <= nums.length <= 10^5`
- `1 <= nums[i], k <= 10^9`

0 1 2 3 4 5 6

2^x 1 $\frac{n(n+1)}{2} - \frac{(n-1)n}{2}$

$n^2 + n - n^2 + n$

$1+2+3+4$

$\frac{n(n+1)}{2} - \frac{n(n-1)}{2}$

$\frac{n(n-1)}{2}$

$3 - (n-1)$

$k = 3$

3 3 3 1 1 2 2 1

$3-2=1$

$n^2 - n - (n^2 - 3n + 2)$

$(n-1)$

$\frac{2n-2}{2}$

$(n-1)$

$\frac{n}{2}(n-1)$

${}^3C_2$

$\frac{(n-1)(n-2)}{2}$

pairs

atleast k pair of indices

$i<j$ & arr[i] == arr[j]

[3, 1, 4, 3, 2, 2, 4] let no. of pairs = n

$i = 0$ [3 → 1] $n = 0$

$i = 1$ [3 → 1, 1 → 1] $n = 0$

$i = 2$ [3 → 1, 1 → 1, 4 → 1] $n = 0$

$i = 3$ [3 → 2, 1 → 1, 4 → 1] $n = 1 < 2$

$i = 4$ [3 → 2, 1 → 1, 4 → 1, 2 → 1] $n = 1 < 2$

ans += (n - i)

$i = 5$ [3 → 2, 1 → 1, 4 → 1, 2 → 2] $n = 2 = 2$

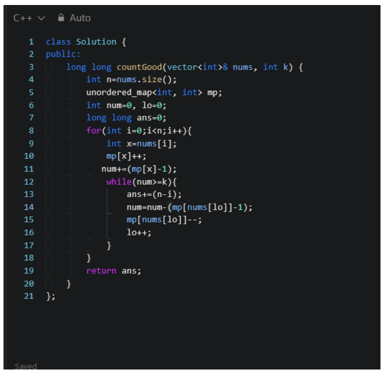
$i = 6$ [3 → 2, 1 → 1, 4 → 2, 2 → 2] $n = 3 > 2$

```
class Solution{
public:
    long long countGood(vector<int>&nums, int k) {
        int n= nums.size();
        unordered_map<int, int> mp;
        int num = 0, lo = 0;
        long long ans = 0;
        for (int i= 0; i<n; i++){
            int x = nums[i];
            mp[x]++;
            num+=(mp[x]-1);
            while(num>=k){
                ans+= (n-i);
                num= num-(mp[nums[lo]]-1);
                lo++; }}
```

return ans++; }};

T.C. O(n)

S.C. O(n)



```cpp
class Solution {
public:
    long long countGood(vector<int>& nums, int k) {
        int n=nums.size();
        unordered_map<int, int> mp;
        int num=0, lo=0;
        long long ans=0;
        for(int i=0;i<n;i++){
            int x=nums[i];
            mp[x]++;
           num+=(mp[x]-1);
            while(num>=k){
                ans+=(n-i);
                num=num-(mp[nums[lo]]-1);
                mp[nums[lo]]--;
                lo++;
            }
        }
        return ans;
    }
};
```