Problem 2537: Count the Number of Good Subarrays

Problem Information

Difficulty: Medium
Acceptance Rate: 65.90%
Paid Only: No
Tags: Array, Hash Table, Sliding Window

Problem Description

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 <= 105` * `1 <= nums[i], k <= 109`

Code Snippets

C++:

```
class Solution {
public:
long long countGood(vector<int>& nums, int k) {

}
};
```

Java:

```
class Solution {
public long countGood(int[] nums, int k) {

}
}
```

Python3:

```
class Solution:
def countGood(self, nums: List[int], k: int) -> int:
```