Problem 3729: Count Distinct Subarrays Divisible by K in Sorted Array

Problem Information

Difficulty: Hard
Acceptance Rate: 26.26%
Paid Only: No
Tags: Array, Hash Table, Prefix Sum

Problem Description

You are given an integer array `nums` **sorted** in **non-descending** order and a positive integer `k`.

A **subarray** of `nums` is **good** if the sum of its elements is **divisible** by `k`.

Return an integer denoting the number of **distinct** **good** subarrays of `nums`.

Subarrays are **distinct** if their sequences of values are. For example, there are 3 **distinct** subarrays in `[1, 1, 1]`, namely `[1]`, `[1, 1]`, and `[1, 1, 1]`.

Example 1:

Input: nums = [1,2,3], k = 3

Output: 3

Explanation:

The good subarrays are `[1, 2]`, `[3]`, and `[1, 2, 3]`. For example, `[1, 2, 3]` is good because the sum of its elements is `1 + 2 + 3 = 6`, and `6 % k = 6 % 3 = 0`.

Example 2:

Input: nums = [2,2,2,2,2,2], k = 6

Output: 2

Explanation:

The good subarrays are `[2, 2, 2]` and `[2, 2, 2, 2, 2, 2]`. For example, `[2, 2, 2]` is good because the sum of its elements is `2 + 2 + 2 = 6`, and `6 % k = 6 % 6 = 0`.

Note that `[2, 2, 2]` is counted only once.

Constraints:

* `1 <= nums.length <= 105` * `1 <= nums[i] <= 109` * `nums` is sorted in non-descending order. * `1 <= k <= 109`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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