Problem 2524: Maximum Frequency Score of a Subarray

Problem Information

Difficulty: Hard
Acceptance Rate: 35.65%
Paid Only: Yes
Tags: Array, Hash Table, Math, Stack, Sliding Window

Problem Description

You are given an integer array `nums` and a **positive** integer `k`.

The **frequency score** of an array is the sum of the **distinct** values in the array raised to the power of their **frequencies** , taking the sum **modulo** `109 + 7`.

* For example, the frequency score of the array `[5,4,5,7,4,4]` is `(43 + 52 + 71) modulo (109 + 7) = 96`.

Return _the**maximum** frequency score of a **subarray** of size _`k` _in_`nums`. You should maximize the value under the modulo and not the actual value.

A **subarray** is a contiguous part of an array.

Example 1:

Input: nums = [1,1,1,2,1,2], k = 3 **Output:** 5 **Explanation:** The subarray [2,1,2] has a frequency score equal to 5. It can be shown that it is the maximum frequency score we can have.

Example 2:

Input: nums = [1,1,1,1,1,1], k = 4 **Output:** 1 **Explanation:** All the subarrays of length 4 have a frequency score equal to 1.

Constraints:

* `1 <= k <= nums.length <= 105` * `1 <= nums[i] <= 106`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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