Problem 3641: Longest Semi-Repeating Subarray

Problem Information

Difficulty: Medium
Acceptance Rate: 66.08%
Paid Only: Yes

Problem Description

You are given an integer array `nums` of length `n` and an integer `k`.

A **semi■repeating** subarray is a contiguous subarray in which at most `k` elements repeat (i.e., appear more than once).

Return the length of the longest **semi■repeating** subarray in `nums`.

Example 1:

Input: nums = [1,2,3,1,2,3,4], k = 2

Output: 6

Explanation:

The longest semi-repeating subarray is `[2, 3, 1, 2, 3, 4]`, which has two repeating elements (2 and 3).

Example 2:

Input: nums = [1,1,1,1,1], k = 4

Output: 5

Explanation:

The longest semi-repeating subarray is `[1, 1, 1, 1, 1]`, which has only one repeating element (1).

Example 3:

Input: nums = [1,1,1,1,1], k = 0

Output: 1

Explanation:

The longest semi-repeating subarray is `[1]`, which has no repeating elements.

Constraints:

* `1 <= nums.length <= 105` * `1 <= nums[i] <= 105` * `0 <= k <= nums.length`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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