Problem 697: Degree of an Array

Problem Information

Difficulty: Easy
Acceptance Rate: 57.90%
Paid Only: No
Tags: Array, Hash Table

Problem Description

Given a non-empty array of non-negative integers `nums`, the **degree** of this array is defined as the maximum frequency of any one of its elements.

Your task is to find the smallest possible length of a (contiguous) subarray of `nums`, that has the same degree as `nums`.

Example 1:

Input: nums = [1,2,2,3,1] **Output:** 2 **Explanation:** The input array has a degree of 2 because both elements 1 and 2 appear twice. Of the subarrays that have the same degree: [1, 2, 2, 3, 1], [1, 2, 2, 3], [2, 2, 3, 1], [1, 2, 2], [2, 2, 3], [2, 2] The shortest length is 2. So return 2.

Example 2:

Input: nums = [1,2,2,3,1,4,2] **Output:** 6 **Explanation:** The degree is 3 because the element 2 is repeated 3 times. So [2,2,3,1,4,2] is the shortest subarray, therefore returning 6.

Constraints:

* `nums.length` will be between 1 and 50,000. * `nums[i]` will be an integer between 0 and 49,999.

Code Snippets

C++:

```
class Solution {
public:
int findShortestSubArray(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public int findShortestSubArray(int[] nums) {

}
}
```

Python3:

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