697. Degree of an Array

Created	@June 15, 2021 7:29 PM
Tags	Easy
link	https://leetcode.com/problems/degree-of-an-array/

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`.

Approach

- Maintain a map for `counts`, `start` and `end` for each integer and another map for `sizes`.
- Iterate through array to fill maps and find `max_frequency` .
- Iterate through map, to check `minimum size` for elements with `max_frequency`

```
class Solution {
public:
    int findShortestSubArray(vector<int>& nums) {
        map<int, pair<int, pair<int, int> > > counts;
        map<int, int> sizes;

        int max_freq = 0;

        for (int i = 0; i < nums.size(); i++) {
            if (counts.find(nums[i]) == counts.end()) {
                counts[nums[i]] = make_pair(1, make_pair(i, i));
                sizes[nums[i]] = 1;
            } else {
                counts[nums[i]].first += 1;
                counts[nums[i]].second.second = i;
                sizes[nums[i]] = counts[nums[i]].second.second - counts[nums[i]].second.first + 1;
            }

            max_freq = max(max_freq, counts[nums[i]].first);
        }

        int min_size = INT_MAX;
        for (auto i: nums) {
            if (counts[i].first == max_freq) {
                min_size = min(min_size, sizes[i]);
            }
        }

        return min_size;
    }
};
```