697 数组的度

label:hash
给定一个非空且只包含非负数的整数数组 nums， 数组的度的定义是指数组里任一元素出现频数的最大值。
你的任务是找到与 nums 拥有相同大小的度的最短连续子数组，返回其长度。
示例 1:
输入: [1, 2, 2, 3, 1]
输出: 2
输入数组的度是2，因为元素1和2的出现频数最大，均为2.
连续子数组里面拥有相同度的有如下所示:
[1, 2, 2, 3, 1], [1, 2, 2, 3], [2, 2, 3, 1], [1, 2, 2], [2, 2, 3], [2, 2]
最短连续子数组[2, 2]的长度为2，所以返回2.

- 以空间为主 map

```
class Solution {
    public int findShortestSubArray(int[] nums) {
        if(null == nums || 0 == nums.length)return 0;
        if(nums.length == 1) return 1;
        // map1统计数组中每个元素出现的次数 比如题中： 1 -> 2 (数组中1出现了2次)
        Map<Integer, Integer> map1 = new HashMap<>();

        // map2记录数组中元素第一次出现时的位置（索引） 比如题中： 1 -> 0 (数组中1第一次出现的位置是0)
        Map<Integer, Integer> map2 = new HashMap<>();

        // map3记录数组相同元素，第一次出现与最后一次出现之间的长度
        // 比如题中：元素1第一次出现的位置是0，最后一次出现的位置是4，它们之间的长度 = 5
        Map<Integer, Integer> map3 = new HashMap<>();  // 一直更新

        for(int i = 0; i < nums.length; i++){
            map1.put(nums[i], map1.getOrDefault(nums[i],0) + 1);
            if(!map2.containsKey(nums[i])){
                map2.put(nums[i],i);
            }
            map3.put(nums[i],i - map2.get(nums[i]) + 1); // 每次更新，因为肯定是越来越大的
        }

        int maxCount = 0;  // 最大度数
        for(int num : map1.values()) maxCount = Math.max(maxCount, num);

        int minLength = Integer.MAX_VALUE;
        for(int key : map1.keySet()){
            if(maxCount == map1.get(key)){  // 最大度可能有多个
                minLength = Math.min(minLength, map3.get(key));
            }
        }
        return minLength;
    }
}
```