**Problem #697: Degree of an Array**

<https://leetcode.com/problems/degree-of-an-array/description/>

**My Solution:**

1. Get the frequency dictionary of nums using python library Counter from collecctions called freqDict.
2. Find the maximum of the values in the frequency dictionary called maxVal.
3. If maximum value is 1, then return 1.
4. Let minInterval be set to infinity.
5. Iterate through the items in the frequency dictionary. If the value is equal to maxVal, then find the index of the first and last occurrence of the key in nums. Then interval will be lastIdx – firstIdx + 1.

If the interval is less than minInterval, update minInterval with Interval.

1. Return minInterval.

from collections import Counter

class Solution:

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

freqDict = Counter(nums)

maxVal = max(freqDict.values())

if maxVal == 1:

return 1

minInterval = math.inf

for k, v in freqDict.items():

if v == maxVal:

firstIdx = nums.index(k)

for j in range(len(nums) - 1, -1, -1):

if nums[j] == k:

lastIdx = j

break

interval = lastIdx - firstIdx + 1

minInterval = min(interval, minInterval)

return minInterval