Problem # 532: K-diff Pairs in an Array (Medium)

-----------------------------------------------------------

<https://leetcode.com/problems/k-diff-pairs-in-an-array/>

My Solution:

<https://leetcode.com/problems/k-diff-pairs-in-an-array/discuss/877426/Simple-Python-3-Solution-using-dictionary-Runtime-beats-99.24>

Simple Python 3 Solution using dictionary -- Runtime beats 99.24%

1. Use Counter from collections to get the frequency of each unique element in nums.
2. If k is 0, then get the number of values in the key, value pair where value > 1. This will give us repeated keys, i.e. (key, key) will form a pair.
3. The keys in the dictionary are unique and basically form a set. Iterate through the list of keys to find whether the corresponding pair element (key + k) is in the list of keys, i.e. we are looking for pairs of the form (key, key + k) in the dictionary and counting them.
4. Return the count.

from collections import Counter

class Solution:

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

my\_dict = Counter(nums)

if k == 0:

count = 0

for value in my\_dict.values():

if value > 1: # key has repeated values, so (key, key) is a pair

count += 1

return(count)

#else: k != 0

keys = my\_dict.keys() # keys in a dictionary are unique i.e. a set

count = 0

for key in keys:

if (k + key) in keys: # (key, k + key) form a pair

count += 1

return(count)