Problem #1338: Reduce array size of the half (Medium)

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

<https://leetcode.com/problems/reduce-array-size-to-the-half/>

My Solution:

<https://leetcode.com/problems/reduce-array-size-to-the-half/discuss/1326236/Simple-Python-3-Solution-Runtime-beats-55.37>

Runtime beats 55.37%

1. Get the frequency of the elements of the array sorted by frequency in reverse order to sorted\_list. Now sorted\_list has tuples with first element being the element from the array and the second element being its frequency (i.e., count of occurrences in arr)>

2. Let half\_size be half the length of arr.

3. Initialize count and total to 0.

4. Iterate through sorted\_list consisting of tuples which are unpacked. The first element of each tuple is the element in arr (i.e., the dictionary key) and the second one is its value (i.e. frequency). Increment count by 1.

Add the value to total.

Check if total is greater than or equal to half\_size.

If so, break.

5. Return the count.

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

from collections import Counter

class Solution:

def minSetSize(self, arr: List[int]) -> int:

sorted\_list = sorted(Counter(arr).items(), key = lambda x:x[1], reverse = True)

half\_size = len(arr)//2

count = 0

total = 0

for x,y in sorted\_list:

total += y

count += 1

if total >= half\_size:

break

return count

=====================================================