Problem # 594: Longest Harmonious Subsequence (Easy)

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

<https://leetcode.com/problems/longest-harmonious-subsequence/>

My Solution:

Runtime beats 85.25%

1. First do the import – from collections import Counter

2. Create a dictionary called my\_dict using Counter with nums to count the frequency of numbers in nums list.

3. Initialize res to 0.

4. Iterate through the keys of my\_dict. If (key + 1) is also present as a key in my\_dict,

then the length of the subsequence will be the frequency of the key and the (key + 1) which we can get from my\_dict. If the length of the subsequence is greater than res, then update res with this length.

5. Finally return res.

from collections import Counter

class Solution:

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

my\_dict = Counter(nums)

res = 0

for key in my\_dict.keys():

if key + 1 in my\_dict.keys():

res = max(res, my\_dict[key] + my\_dict[key+1])

return res