Problem # 228: Summary Ranges (Easy)

<https://leetcode.com/problems/summary-ranges/>

My Solution:

<https://leetcode.com/problems/summary-ranges/discuss/914548/Simple-Python-3-Solution-Runtime-beats-90.41>

1. Let n be the length of nums array.

If n is equal to 0, return an empty set.

If n is equal to 1, return the element of nums converted to string in a list.

2. Initailize result list to an empty list. Initalize temp\_list to the first element of nums i.e. nums[0].

3. Iterate from 1 through n.

If the value of nums at index i is equal to 1 + the value of nums at index (i – 1), that is nums[i] is next integer after nums[I – 1], then append the value of nums at index i to the temp\_list.

4. Otherwise, check the length of the temp\_list. If the length of temp\_list is 1, then append the element of temp\_list converted to string to the result list. If the length of the temp\_list is greater than 1, then create a string with first element of temp\_list, ‘->’, and last element of temp\_list. Append the string created to the result list. Set temp\_list to a list with the value of index i of nums.

5. When done with iterating through the nums array, if temp\_list has length of 1, then append to result list the element of the temp\_list converted to a string,. Otherwise, if the length of temp\_list is more than 1, then create a string with first element of temp\_list, ‘->’, and last element of temp\_list. Append the string created to the result list.

6. Return the result list.

class Solution:

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

n = len(nums)

if n == 0:

return []

elif n == 1:

return [str(nums[0])]

result = []

temp\_list = [nums[0]]

for i in range(1, n):

if nums[i] == nums[i - 1] + 1:

temp\_list.append(nums[i])

else:

if len(temp\_list) == 1:

result.append(str(temp\_list[0]))

else:

result.append(str(temp\_list[0]) + "->" + str(temp\_list[-1]))

temp\_list = [nums[i]]

if len(temp\_list) == 1:

result.append(str(temp\_list[0]))

else:

result.append(str(temp\_list[0]) + "->" + str(temp\_list[-1]))

return result