**Problem #443: String Compression**

<https://leetcode.com/problems/string-compression/description/>

**My Solution:**

1. Let lastChar be the element at index 0 in chars.
2. Initialize count to 1
3. Initialize read pointer j to 0.
4. Traverse chars and read each character. If the current character is the same as previous one, just increment count by 1. Otherwise, write the character at index j as lastChar and increment j by 1.

If the count is between 2 and 9 (both inclusive), then convert count to a string and write it into character at index j. Increment j.

If count is greater than or equal to 10, the let num\_str be count converted to a string.

Write this string into chars with j pointer.

Reset count to 1.

Reset lastChar to character at index I.

Write lastChar into the character at index j.

1. At the end of the traversal, write the last character into chars. Then write its count into chars.
2. Make chars to be the first j character
3. Return length of chars

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

class Solution:

def compress(self, chars: List[str]) -> int:

lastChar = chars[0]

count = 1

j = 0

for i in range(1, len(chars)):

if chars[i] == lastChar:

count += 1

else: # chars[i] != lastChar:

chars[j] = lastChar

j += 1

if count > 1 and count < 10:

chars[j] = str(count)

j += 1

elif count >= 10:

num\_str = str(count)

for k in range(len(num\_str)):

chars[j] = num\_str[k]

j += 1

count = 1

lastChar = chars[i]

chars[j] = lastChar

#print("i = ", i, "j = ", j, "chars = ", chars, "lastChar = ", lastChar, "count = ", count )

# Update last char at the end-- update the last char

chars[j] = lastChar

j += 1

if count > 1 and count < 10:

chars[j] = str(count)

j += 1

elif count >= 10:

num\_str = str(count)

for k in range(len(num\_str)):

chars[j] = num\_str[k]

j += 1

chars = chars[0 : j]

return len(chars)