[443. String Compression](https://leetcode.com/problems/string-compression/)

Solved

Medium

Topics

Companies

Hint

Given an array of characters chars, compress it using the following algorithm:

Begin with an empty string s. For each group of **consecutive repeating characters** in chars:

* If the group's length is 1, append the character to s.
* Otherwise, append the character followed by the group's length.

The compressed string s **should not be returned separately**, but instead, be stored **in the input character array chars**. Note that group lengths that are 10 or longer will be split into multiple characters in chars.

After you are done **modifying the input array,** return *the new length of the array*.

You must write an algorithm that uses only constant extra space.

**Example 1:**

**Input:** chars = ["a","a","b","b","c","c","c"]

**Output:** Return 6, and the first 6 characters of the input array should be: ["a","2","b","2","c","3"]

**Explanation:** The groups are "aa", "bb", and "ccc". This compresses to "a2b2c3".

**Example 2:**

**Input:** chars = ["a"]

**Output:** Return 1, and the first character of the input array should be: ["a"]

**Explanation:** The only group is "a", which remains uncompressed since it's a single character.

**Example 3:**

**Input:** chars = ["a","b","b","b","b","b","b","b","b","b","b","b","b"]

**Output:** Return 4, and the first 4 characters of the input array should be: ["a","b","1","2"].

**Explanation:** The groups are "a" and "bbbbbbbbbbbb". This compresses to "ab12".

**Constraints:**

* 1 <= chars.length <= 2000
* chars[i] is a lowercase English letter, uppercase English letter, digit, or symbol.

Solution

What does problem say

Problem says that we need to compress the string that given to us

What how to compress the string

For example

chars = ["a","a","b","b","c","c","c"]

for given example we can compress it to like a2b2c3

and in the problem says to return length of the compressed string in this example it would be 6

key observations

1. we need to return the length of the compressed length
2. if the occurrence of a character is more than two digits in the case of 2 or 3 input given in the problem then we have to count it in separated way

for example for a character has as a count of 10

it should be 1 , 0 in the string

then it would be a,1,0

intuition to solve this problem is to

we traverse through the character array , for each i char in array we try to count the occurrence of the that I character and using idx variable we assign the character with occurrences increment I and idx and count =0 for other character and

if count is greater than single digit then we have to make it string and traverse it and assign it to character array   
in end we have to return idx which is length of the compressed string

time comp – first loop for traverse the char array 🡪 O(n)

second loop for char occurrence which can be 🡪 O(count)

if count is greater than single digit

then loop through the length of count 🡪 O(len(count))

we can say it would be O(n)

we can say space is constant we are using input array

but it should be O(n)

we can solve this problem in linear time and space

class Solution {

    public int compress(char[] chars) {

        int idx = 0;

        int i = 0;

        while (i < chars.length) {

            char ch = chars[i];

            int count = 0;

            while (i < chars.length && chars[i] == ch) {

                count++;

                i++;

            }

            chars[idx++] = ch;

            if (count > 1) {

                String s = String.valueOf(count);

                for (char c : s.toCharArray()) {

                    chars[idx++] = c;

                }

            }

        }

        return idx;

    }