

1.6 String Compression - Write a function to compress a string using char counts $aaabbbccdd \rightarrow a3b2c1d2$

Parameters (String: compress)

Return (compressed string or original)

Simplest Problem

IN "a"

OUT "a"

V2

IN ""

OUT ""

V3

IN \rightarrow "aa'bb"

OUT \rightarrow ~~"aa'bb"~~ \rightarrow "aab"

IN \rightarrow "aaaccbbbdcc"

OUT \rightarrow "a3c2b3d1c1"

Patterns & Brainstorm

- Only check if there is more than one occurrence in a row.
- Use a flag

Solutions

Separate array
- If curr == prev
inc count
flag append char
else
append count
count = 0
append cur

Time Space
 $O(n)$ $O(n)$

Notes

- Final count was not added because loop terminated.
- When ever you check a previous element \rightarrow not this
- Never call strings 'str'
- Never add two strings, append them separately to a list.
- Don't use extend unless you already have a list

Code

```
def compress(str):  
    if len(str) <= 2:  
        return str  
    comp = []  
    count = 1  
    for i in range(len(str)):  
        if i == 0:  
            continue  
        if str[i] == str[i-1]:  
            count += 1  
        else:  
            count = 1  
            comp.append(str[i-1])  
            comp.append(count)
```

```
if (len(str) < len(comp)):  
    return str  
return comp
```