


▲

hjt486★ 13Last Edit: November 21, 2019 12:01 PM365 VIEWS

11

This is quite self-explanatory, but I'll try my best to explain, as you can see from the example:  
S = "aaaba", For every continuous letters like 'aaa', you can calculate as:

'a' for 3 times,  
'aa' for 2 times,  
'a' for 1 time.

so you only need to know how many of these continuous letters are, and then for each of them, calculate the sum by adding from 1 to the length of the continuous letters, for 'aaa' it's 1+2+3=6. In our example, there are 'aaa', 'b' and 'a', so it's (1+2+3) + 1 + 1.

At the beginning of our code, to make the code easier to write without considering too many edge cases, I concatenate two spaces at the beginning and the end of the string.


Then we have two variables, one as 'total' to hold the sum, and one 'count' as the counter for the continuous letters. If we found i-th letter is not same as the i-1th letter, we reset the counter to 1, otherwise they're the same, we add 1 to the counter. For each time, we simply add the counter value to the total, then at the end we return 'total'.

```
class Solution:
    def countLetters(self, S: str) -> int:
        S = ' ' + S + ' '
        total, count = 0, 1
        for i in range(1, len(S)-1):
            if S[i] != S[i-1]:
                count = 1
            else:
                count += 1
            total += count
        return total
```

python 3

Type comment here... (Markdown is supported)

Post

RajatThosar★ 2January 10, 2020 5:09 AM

Had the same idea. Here's my solution:


```
def countLetters(self, S: str) -> int:
    if not S:
        return 0


    prev_char = S[0]
    count = 0
    result = 0

    for char in S:
```

Read More


▲ 1 ▼


Reply

alankritjoshi★ 18November 24, 2019 5:36 AM

you are a genius!


▲ 1 ▼

Reply

tracyxu1993★ 46November 21, 2019 9:33 PM

Genius!

▲ 0 ▼

Reply