Back | Python3 beats 100%

justin801514 ★ 46 Last Edit: August 26, 2019 2:24 AM 1.3K VIEWS

7

```
class Solution:
    def calculateTime(self, keyboard: str, word: str) -> int:
        cur_index = 0
        time = 0
        for w in word:
            next_index = keyboard.index(w)
            time += abs(next_index - cur_index)
            cur_index = next_index
        return time
```

Comments: 5 Best Most Votes Newest to Oldest Oldest to Newest

Type comment here... (Markdown is supported)

The keyboard.index(w) takes worstcase o(N) time. You'll be doing it on all the characters in a word

6 Show 3 replies Reply

Padampalle ★ 12 September 26, 2019 12:45 AM

Should have hashed the key and its index. like this: {'a':0, 'b':1....}
so you don't need to find the location again and again.

4 Show 2 replies Reply

Xyzzy123 ★ 51 April 9, 2020 11:19 PM

Here's an alternative that uses a dictionary for faster O(1) lookups of distance:

```
def calculateTime(self, keyboard: str, word: str) -> int:

    dist = {ch: idx for idx, ch in enumerate(keyboard)}

    time = 0

    for i, ch in enumerate(word):
        if i == 0:
            time += dist[ch]
```

Read More

1 Reply

planetzjy ★ 0 February 20, 2020 1:57 AM

Memory usage maybe, but not run time.
Would be faster if index info is hashed like Padampalle said.

0 Reply

ksamir ★ 0 August 27, 2019 5:20 PM

Beautiful!

0 Reply