Average Rating: 4.29 (55 votes)

Dec. 9, 2018 | 278.9K views

387. First Unique Character in a String decided

Examples:

```
s = "leetcode"
return 0.
s = "loveleetcode"
return 2.
```

Given a string, find the first non-repeating character in it and return its index. If it doesn't exist, return -1.

Note: You may assume the string contains only lowercase English letters.

Solution

The best possible solution here could be of a linear time because to ensure that the character is unique you

a character is unique or not.

Approach 1: Linear time solution

have to check the whole string anyway. The idea is to go through the string and save in a hash map the number of times each character appears in

the string. That would take $\mathcal{O}(N)$ time, where N is a number of characters in the string. And then we go through the string the second time, this time we use the hash map as a reference to check if

If the character is unique, one could just return its index. The complexity of the second iteration is $\mathcal{O}(N)$ as well.

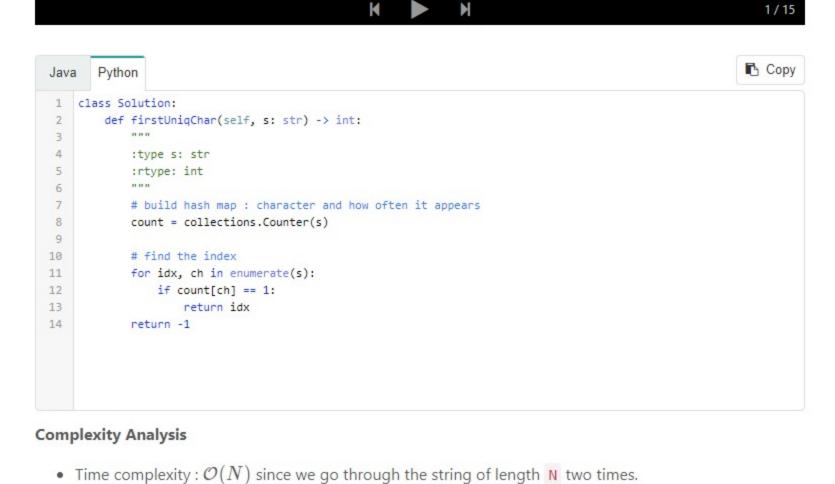
LOVELEETCODE

Hash map



Build L:1





• Space complexity : $\mathcal{O}(1)$ because English alphabet contains 26 letters.



public int firstUniqChar(String s) { int [] charCount = new int[26];

8 A V E Share Reply

(12345678)

SHOW 4 REPLIES

Read More