387 First Unique Character in a String

2018年4月10日　14:40

Question：

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

Examples:

s = "leetcode"

return 0.

s = "loveleetcode",

return 2.

Note: You may assume the string contain only lowercase letters.

来自 <https://leetcode.com/problems/first-unique-character-in-a-string/description/>

给定一个字符串，找到它的第一个不重复的字符，并返回它的索引。如果不存在，则返回 -1。

案例:

s = "leetcode"

返回 0.

s = "loveleetcode",

返回 2.

Solution for Python3:

```python
class Solution1:
    def firstUniqChar(self, s):
        """
        :type s: str
        :rtype: int
        """
        from collections import OrderedDict
        d = OrderedDict()
        for i in range(len(s)):
            if d.get(s[i]):
                d[s[i]] += 1
            else:
                d[s[i]] = 1
        print(d)
        if 1 not in d.values():
            return -1
        return s.find(list(d.keys())[list(d.values()).index(1)])

class Solution2:
    def firstUniqChar(self, s):
        """
        :type s: str
        :rtype: int
        """
```

```python
            letters='abcdefghijklmnopqrstuvwxyz'
            index = [s.index(l) for l in letters if s.count(l) == 1]
            return min(index) if len(index) > 0 else -1

class Solution3:
    def firstUniqChar(self, s):
        """
        :type s: str
        :rtype: int
        """
        letters = [0] * 26
        for i in s:
            letters[ord(i) - ord('a')] += 1
        for i in range(len(s)):
            if letters[ord(s[i]) - ord('a')] == 1:
                return i
        return -1
```

Solution for C++:

```cpp
class Solution {
public:
    int firstUniqChar(string s) {
        int array[26] = {0};
        for (char c : s) {
            array[c - 'a']++;
        }
        for (int i = 0; i < s.legnth(); i++) {
            if (array[s[i] - 'a'] == 1) {
                return i;
            }
        }
        return -1;
    }
};
```