**Problem #387: First Unique Character in a String. (Easy)**

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

Given a string s, *find the first non-repeating character in it and return its index*. If it does not exist, return -1.

**Example 1:**

**Input:** s = "leetcode"

**Output:** 0

**Example 2:**

**Input:** s = "loveleetcode"

**Output:** 2

**Example 3:**

**Input:** s = "aabb"

**Output:** -1

* **Constraints:**
* 1 <= s.length <= 105
* s consists of only lowercase English letters.

**My Solution:**

1. Use an ordered dictionary to store the letters in the string as keys and number of occurrences of each key is the value.
2. Iterate through the dictionary to find that first key with value 1 and return its index.
3. If no key is found with value of 1, then return -1

from collections import OrderedDict

class Solution:

def firstUniqChar(self, s: str) -> int:

od = OrderedDict()

for i in range(len(s)):

if s[i] in od:

od[s[i]] += 1

else:

od[s[i]] = 1

for k, v in od.items():

if v == 1:

return s.index(k)

return -1