First Unique Character in a String

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.

Solution:

```
class Solution {
    public int firstUniqChar(String s) {
        if(s.isEmpty()){
            return -1;
        }
        Map<Character,Integer> count = new HashMap<>();
        for( int i = 0; i < s.length(); i++) {
            count.put(s.charAt(i), count.getOrDefault( s.charAt(i), 0) + 1);
        }

        for( int i =0; i< s.length(); i++){
            if(count.containsKey(s.charAt(i))){
                if(count.get(s.charAt(i)) == 1)
                   {
                       return i;
                   }
            }
        }
        return -1;


    }
}
```