```
/*
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.



注意事项：您可以假定该字符串只包含小写字母。

来源：力扣（LeetCode）
链接：https://leetcode-cn.com/problems/first-unique-character-in-a-string
著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
*/
```

---

分析：

- 首先想到的是能不能利用异或操作的特性来筛选，发现不是太好利用(唯一的字符并不是只有一个，不重复的话字符也不一定是2个)
- 硬干法，第一次遍历获得计数各个小写字母出现的次数，第二次遍历筛选出第一个只出现了一次的小写字母在字符串的位置索引，见方法一.
- 方法一: 额外空间$O(26)$即常数空间复杂度, 时间复杂度,第一次遍历$O(N)$,第二次遍历最差$O(N)$,加起来还是$O(N)$,其中$N$是输入字符串的长度.

---

方法一:C_Solution,两次遍历.

```c
int firstUniqChar(char * s){

    int char_num[26]    = {0,};
    int i               = 0;
    int ret_val         = -1;

    memset(char_num,0,sizeof(int)*26);

    /*第一次遍历,$O(N)$*/
    while(s[i])
    {
        ++char_num[s[i]-'a'];
        ++i;
    }

    /*第二次遍历最差$O(N)$*/
```

```
    i = 0;
    while(s[i])
    {
        if(char_num[s[i]-'a']==1)
        {
            ret_val = i;
            break;
        }
        ++i;
    }
    return ret_val;
}
/*
执行结果：通过
显示详情
执行用时 :16 ms, 在所有 C 提交中击败了84.11%的用户
内存消耗 :8.1 MB, 在所有 C 提交中击败了94.98%的用户
*/
```

---

AlimyBreak
2019.07.16

---