



链滴

第一个只出现一次的字符

作者: [yudake](#)

原文链接: <https://ld246.com/article/1518871934677>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

题目描述

在一个字符串($1 \leq \text{字符串长度} \leq 10000$, 全部由字母组成)中找到第一个只出现一次的字符,并返回的位置

解题思路

两种方法:

- 利用两个链表或者一个map;
- 因为全部由字母组成, 可以利用一个大小为52的数组记录字符的出现情况 (为了拓展性, 这里用了小为256的数组):
 - 数组初始化为-1;
 - 如果数组第一次出现, 则利用字符的ASCII大小作为数组的index, 数组相应位置的值设置为字符在字符串的位置;
 - 如果数组内相应位置不为-1, 说明字符已经出现过, 把数组相应位置设为-2;
 - 最后再遍历一遍数组, 找到非负数的最小值, 就是所求的位置。

代码

代码一:

```
import java.util.HashMap;
public class Solution {
    public int FirstNotRepeatingChar(String str) {
        if (str == null || str.trim().length() == 0)
            return -1;
        HashMap<Character, Integer> map = new HashMap<>();
        for (int i = 0; i < str.length(); i++) {
            if (map.containsKey(str.charAt(i)))
                map.put(str.charAt(i), map.get(str.charAt(i))+1);
            else
                map.put(str.charAt(i), 1);
        }
        for (int i = 0; i < str.length(); i++) {
            if (map.get(str.charAt(i)) == 1)
                return i;
        }
        return -1;
    }
}
```

代码二:

```
public class Solution {
    public int FirstNotRepeatingChar(String str) {
        if (str == null || str.length() == 0)
            return -1;
        int[] tmp = new int[256];
```

```
for (int i = 0; i < 256; i++) {
    tmp[i] = -1;
}
for (int i = 0; i < str.length(); i++) {
    char c = str.charAt(i);
    if (tmp[c] == -1)
        tmp[c] = i;
    else if (tmp[c] > -1)
        tmp[c] = -2;
}
int ret = Integer.MAX_VALUE;
for (int i = 0; i < 256; i++) {
    if (tmp[i] > -1)
        ret = ret < tmp[i] ? ret : tmp[i];
}
return ret;
}
}
```