

```

1  /*
2  给定一个字符串，请你找出其中不含有重复字符的 最长子串 的长度。
3
4  示例 1:
5
6  输入: "abcabcbb"
7  输出: 3
8  解释: 因为无重复字符的最长子串是 "abc"，所以其长度为 3。
9
10 示例 2:
11
12 输入: "bbbbb"
13 输出: 1
14 解释: 因为无重复字符的最长子串是 "b"，所以其长度为 1。
15
16 示例 3:
17
18 输入: "pwwkew"
19 输出: 3
20 解释: 因为无重复字符的最长子串是 "wke"，所以其长度为 3。
21      请注意，你的答案必须是 子串 的长度，"pwke" 是一个子序列，不是子串。
22
23 来源：力扣（LeetCode）
24 链接：https://leetcode-cn.com/problems/longest-substring-without-repeating-characters
25 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。
26 */

```

分析:

- 本题与[485](#)解法类似,区别在于需要增加一个函数用于判断当前索引到的值是否在跟当前窗口中的元素是有重复.

方法一:C_双指针窗口滑动法

```

1  /*****
2  若target在 s[start,end)中没有重复的就返回-1,
3  若有重复的就返回重复数据所在的索引值.
4  *****/
5  int exsit_double( char* s,
6                   int start,
7                   int end,
8                   char target
9                   )
10 {
11     int ret_val = -1;
12     int i = 0;
13
14
15     for(i = start; i < end; i++)
16     {
17         if(target==s[i])
18         {

```

```

19         ret_val = i;
20         break;
21     }
22 }
23 return ret_val;
24 }
25
26 int lengthOfLongestSubstring(char * s)
27 {
28
29     int ind_front    = 0    ;
30     int ind_end      = 1    ;
31     int ret_val      = 0    ;
32     int temp         = 0    ;
33
34     if(s[0])        /* 原始数据长度不为0*/
35     {
36         while(s[ind_end])
37         {
38             temp = exsit_double(s,ind_front,ind_end,s[ind_end]);
39             if(temp== -1)
40             {
41                 ind_end++;
42             }
43             else
44             {
45                 ret_val = ret_val >= (ind_end-ind_front) ? ret_val :
(ind_end-ind_front) ;
46                 ind_front = temp + 1
;
47                 ind_end++;
;
48             }
49         }
50         ret_val = ret_val >= (ind_end-ind_front) ? ret_val : (ind_end-
ind_front) ;
51     }
52     return ret_val;
53 }
54 /*
55 执行结果:
56 通过
57 显示详情
58 执行用时 :8 ms, 在所有 C 提交中击败了79.84% 的用户
59 内存消耗 :7.2 MB, 在所有 C 提交中击败了75.45%的用户
60 */

```