

无重复字符的最长子串

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。

示例 1:

输入: "abcabcbb"

输出: 3

解释: 因为无重复字符的最长子串是 "abc", 所以其长度为 3。

示例 2:

输入: "bbbbbb"

输出: 1

解释: 因为无重复字符的最长子串是 "b", 所以其长度为 1。

示例 3:

输入: "pwwkew"

输出: 3

解释: 因为无重复字符的最长子串是 "wke", 所以其长度为 3。

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。

解题思路:

设立左指针 a 和右指针 b

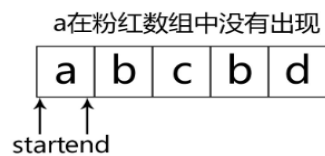
b 指针向右侧伸缩{

对每个 A[b]判断是否在之前的数组出现过;

如果出现，指针 a 指向出现过的位置的下一个位置;更新右指针和最大长度;

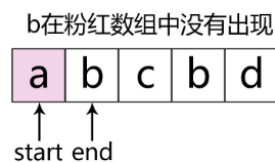
}

无重复字符的最长子串



maxLength:0

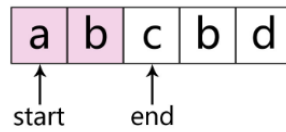
无重复字符的最长子串



maxLength:1

无重复字符的最长字串

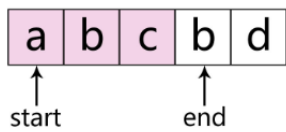
c在粉红数组中没有出现



maxLength:2

无重复字符的最长字串

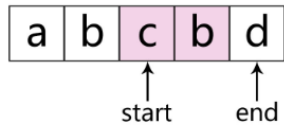
b在粉红数组中出现，修改start值为出现过位置之后的位置



maxLength:3

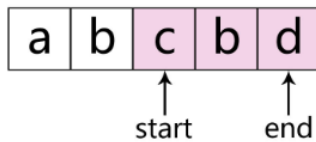
无重复字符的最长字串

d在粉红数组中没有出现



maxLength:3

无重复字符的最长字串



maxLength:3

```
class Solution {  
public:  
    int lengthOfLongestSubstring(string s) {
```

```
//滑动窗口
int start=0,end=0;
int len=0,res=0;
while(end<s.size())
{
    char tmp=s[end];//先将 end 暂存 方便与滑窗里比较
    for(int i=start;i<end;i++)
    {
        if(tmp==s[i])//存在相同的
        {
            start=i+1;//左边界,跳过相同数字,移动至相同数字的下一位置处
            len=end-start;
            break;
        }
    }
    end++;//右边界搜索
    len++;
    res=max(res,len);//找到长度最长的子串
}
return res;
}
};
```