

## 一、题目

Longest Substring Without Repeating Characters, 具体请自行搜索。

这个题目, 我看了一下, 经过一番思考, 我觉得实现起来不是很复杂。

但要做到bug free有点难度, 主要是边界的问题。

二、这个题目, 我自己实现, 没有参考代码

提交了5次:

第1次: Wrong Answer, 主要是" "这个空串不对

第2次、第3次: Runtime Error,"**au**" out of Range

第4次: Wrong Answer, 主要是dvdf 不对

第5次: 终于对了

## 三、改进

虽然自己实现了该问题, 但边界的考虑不周, 总共错误了4次。

目前存在的问题, 就是性能不够好。

看看其他人的实现, 再写3次。

争取做到bug free, 性能足够好。

下面是我的完整代码实现, 需要的拿去:

```
#include<iostream>
using namespace std;

class Solution{
public:
    int lengthOfLongestSubstring(string s){
        int curr = 0, len = s.length();
        int start, end, maxLength=0;
        if(len == 1){
            maxLength = 1;
            return maxLength;
        }
        while(curr<len){
            start = curr;
            end = start + 1;
            if(end>=len){
                break;
            }else{
                string sub = s.substr(start, end-start);
                while(end<len && sub.find(s.at(end)) == -1 ){
                    if(end<=len){
                        end++;
                        sub = s.substr(start, end-start);
                    }else{
                        break;
                    }
                }
            }
        }
    }
};
```

```

        }
        if(maxLength<(end-start)){
            maxLength = end - start;
        }
    }
    curr++;
}

return maxLength;
}
};

int main(){
    Solution s;
    cout<<s.lengthOfLongestSubstring("abcabcbb")<<endl;
    cout<<s.lengthOfLongestSubstring("bbbbbb")<<endl;
    cout<<s.lengthOfLongestSubstring("pwwkew")<<endl;
    cout<<s.lengthOfLongestSubstring(" ")<<endl;
    cout<<s.lengthOfLongestSubstring("  ")<<endl;
    cout<<s.lengthOfLongestSubstring("")<<endl;
    cout<<s.lengthOfLongestSubstring("au")<<endl;
    cout<<s.lengthOfLongestSubstring("dvd")<<endl;
    return 0;
}

```

网上找一个性能稍微好点也容易理解的:

```

#include<iostream>
#include<vector>
using namespace std;

class Solution{
public:    //pwwkew
    int lengthOfLongestSubstring(string s){
        int res = 0;
        vector<int> m(128,0);
        for(int i=0,j=0;j<s.size();j++){
            if(m[s[j]]++ == 0){
                res = max(res,j-i+1);
            }else{
                while(i<j && m[s[j]]>1){
                    m[s[i]]--;
                    i++;
                }
            }
        }
        return res;
    }
};

int main(){
    Solution s;
    cout<<s.lengthOfLongestSubstring("abcabcbb")<<endl;
    cout<<s.lengthOfLongestSubstring("bbbbbb")<<endl;
    cout<<s.lengthOfLongestSubstring("pwwkew")<<endl;
    cout<<s.lengthOfLongestSubstring(" ")<<endl;
    cout<<s.lengthOfLongestSubstring("  ")<<endl;

```

```
cout<<s.lengthOfLongestSubstring("")<<endl;  
cout<<s.lengthOfLongestSubstring("au")<<endl;  
cout<<s.lengthOfLongestSubstring("dvd")<<endl;  
return 0;  
}
```