

3. Longest Substring Without Repeating Characters

题目描述: <https://leetcode.com/problems/longest-substring-without-repeating-characters/>

给定一个字符串，求这个字符串满足如下要求的子串：

1. 没有重复字母
2. 在所有子串中是最长的

例如：

```
"abcabcbb", "abc", the length is 3.  
"bbbbbb", "b", the length is 1.  
"pwwkew", "wke", the length of 3.
```

解题思路：

1. 使用两个pointer，begin和end记录子串的开始和结束位置，都从头开始
2. 每当遇到一个不重复的字符就end++
3. 如果遇到重复的字符，例如为“a”，取当前长度和之前结果的最大值，end++，begin移动到当前子串的“a”后面。
4. 以此类推。

因此我们需要记录的有：当前子串中都有哪些字符，这些字符的位置都在哪儿。

代码：

```

class Solution {
public:
    int lengthOfLongestSubstring(string s) {
        int c[177] = {0};
        int begin = 0, end = 0, M = 0;
        string res;
        while(end < s.size() && begin <= end){
            char now = s.at(end);
            if(c[now] == 0){
                c[now] = ++end;
            }
            else{
                M = max(M, end - begin);
                if(M == end-begin)
                    res = ""+s.substr(begin, end-begin);
                int endloc = c[now];
                for(int i = begin; i < endloc; i++){
                    char del = s.at(i);
                    c[del] = 0;
                    if(del == now){
                        c[now] = ++end;
                        begin = i + 1;
                        break;
                    }
                }
            }
        }
        return max(M, end-begin);
    }
};

```