Longest Substring Without Repeating Characters

Given a string, find the length of the **longest substring** without repeating characters.

**Example 1:**

Input: "abcabcbb"  
Output: 3   
Explanation: The answer is "abc", with the length of 3.

**Example 2:**

Input: "bbbbb"  
Output: 1  
Explanation: The answer is "b", with the length of 1.

**Example 3:**

Input: "pwwkew"  
Output: 3  
Explanation: The answer is "wke", with the length of 3.   
 Note that the answer must be a substring, "pwke" is a subsequence and not a substring.

**解法1**

暴力法。枚举每个子串，判断是否重复。答案会超时

int lengthOfLongestSubstring(string s) {  
 if(s.size() == 0)return 0;  
 int maxlen = 1;  
 for(int i = 0; i < s.size(); ++i){  
 for(int j = i + 1; j < s.size(); ++j){  
 map<char, bool>mp;  
 bool flag = true;  
 for(int k = i; k <= j; ++k){  
 if(mp[s[k]] == false)mp[s[k]] = true;  
 else{  
 flag = false;  
 break;  
 }  
 }  
 if(flag)maxlen = max(maxlen, j - i + 1);  
 }  
 }  
 return maxlen;  
}

**解法2**

滑动窗口

int lengthOfLongestSubstring(string s) {  
 if(s.size() == 0)return 0;  
 int maxlen = 1, preLen = 1;  
 for(int i = 1; i < s.size(); ++i){  
 int pos = i - preLen - 1;  
 for(int j = pos + 1; j < i; ++j){  
 if(s[j] == s[i]){  
 pos = j;  
 }  
 }  
 preLen = i - pos;  
 maxlen = max(maxlen, preLen);  
 }  
 return maxlen;  
}