

问题：给定一个字符串，找到最长的无重复的连续子串。

Examples:

Given "abcabcbb", the answer is "abc", which the length is 3.

Given "bbbbbb", the answer is "b", with the length of 1.

Given "pwwkew", 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.

```
public class L3 {  
    //这是一个动态规划问题，即找出每个字符前的最大不重复子串长度，然后找到最长的那个  
    public int lengthOfLongestSubstring(String s) {  
        int left = 0;  
        int right = 0;  
        int size = 0;  
        HashSet<Character> hashSet = new HashSet<Character>();  
        while(right < s.length()) {  
            if(!hashSet.contains(s.charAt(right))) { //如果hashSet中不包含此字符，则将此字符加入hashSet中，然后比较得到目前为止最长的那个  
                hashSet.add(s.charAt(right++));  
                size = Math.max(size, hashSet.size());  
            }else {  
                hashSet.remove(s.charAt(left++)); //如果hashSet中包含此字符，则一直删除一直删除，直到hashSet中不包含此字符  
            }  
        }  
        return size;  
    }  
}
```