3. Longest Substring Without Repeating Characters

Medium

4653233FavoriteShare

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.

Java:

class Solution {

public int lengthOfLongestSubstring(String s) {

int len=s.length();

if(len==0) return 0;

HashMap<Character,Integer> hm = new HashMap<Character,Integer>();

int i=0,max=0,min=0;

while(i<len){

if(hm.get(s.charAt(i))==null) {

hm.put(s.charAt(i),i);

max=i-min>max?i-min:max;

}else{

min=hm.get(s.charAt(i))+1>min?hm.get(s.charAt(i))+1:min;

hm.put(s.charAt(i),i);

max=i-(min)>max?i-(min):max;

}

i++;

}

return max+1;

}

}

Success

[Details](https://leetcode.com/submissions/detail/204544284/)

Runtime: 25 ms, faster than 79.63% of Java online submissions for Longest Substring Without Repeating Characters.

C:

int lengthOfLongestSubstring(char\* s) {

if(s[0]=='\0') return 0;

else if(s[1]=='\0') return 1;

int i=1,left=0,max=0;

while(s[i]!='\0'){

for(int k=left;k<i;k++){

if(s[k]==s[i]){

left=k+1;

max=(i-left)>max?i-left:max;

break;

}

}

max=(i-left)>max?i-left:max;

i++;

}

return max+1;

}

Success

[Details](https://leetcode.com/submissions/detail/204549335/)

Runtime: 8 ms, faster than 100.00% of C online submissions for Longest Substring Without Repeating Characters.