

## Longest Substring Without Repeating Characters

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

### Example 1:

**Input:** `s = "abcabcbb"`

**Output:** `3`

**Explanation:** The answer is "abc", with the length of 3.

### Example 2:

**Input:** `s = "bbbbbb"`

**Output:** `1`

**Explanation:** The answer is "b", with the length of 1.

### Example 3:

**Input:** `s = "pwwkew"`

**Output:** `3`

**Explanation:** The answer is "wke", with the length of 3.

Notice that the answer must be a substring, "pwke" is a subsequence and not a substring.

### Example 4:

**Input:** `s = ""`

**Output:** `0`

## SOLUTION

```
public class Solution {  
  
    public int lengthOfLongestSubstring(String A) {  
  
        int result = 0;  
  
        int start = 0;
```

```
int end = 0;

int strLen = A.length();

Set<Character> st = new HashSet<>();

while(start < strLen && end < strLen) {

    if(!st.contains(A.charAt(end))) {

        st.add(A.charAt(end++));

        result = Math.max(result, end-start);

    } else {

        st.remove(A.charAt(start++));

    }

}

return result;

}
```

**Output**

Your previous code was restored from your local storage. [Reset to default](#)

Testcase Run Code Result Debugger

**Accepted** Runtime: 0 ms

Your input "abcabcbb"

Output 3

Expected 3

Diff

1/241

Next >

Console

[Use Example Testcases](#)



Run Code

Submit



ENG

22:14  
22-06-2021

