3. Longest Substring Without Repeating Characters

Medium 👍 12671 👎 670 ♡ Add to List ⎙ Share

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 = "bbbbb"
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
```

Constraints:

- `0 <= s.length <= 5 * 10`^{4}
- `s` consists of English letters, digits, symbols and spaces.

While add right, update window counter.
When window not satisified, shrink window.
While add left, reduce window counter.
When window satisfied, update the final result.

```cpp
class Solution {
public:
    int lengthOfLongestSubstring(string s) {
        unordered_map<char, int> window;

        int left = 0, right = 0;
        bool valid = true;

        int length = 0;

        while (right < s.size()) {
            // c is the char adding to window
            char c = s[right];
            // move right side of window
            right++;
            // update window counter and valid
            window[c]++;
            if (window[c] > 1) {
                valid = false;
            }

            while (!valid) {
                // d is the char removing from window
                char d = s[left];
                // move left side of window
                left++;
                // update window counter and valid
                window[d]--;
                if (window[d] == 1) {
                    valid = true;
                }
            }

            // update the final result
            if (length < right - left) {
                length = right - left;
            }

        }

        return length;
    }
};
```