Problem 3: Longest Substring Without Repeating Characters

Problem Information

Difficulty: Medium
Acceptance Rate: 37.91%
Paid Only: No
Tags: Hash Table, String, Sliding Window

Problem Description

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

Example 1:

Input: s = "abcabcbb" **Output:** 3 **Explanation:** The answer is "abc", with the length of 3. Note that "bca" and "cab" are also correct answers.

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.

Constraints:

* `0 <= s.length <= 5 * 104` * `s` consists of English letters, digits, symbols and spaces.

Code Snippets

C++:

```
class Solution {
public:
int lengthOfLongestSubstring(string s) {

}
};
```

Java:

```
class Solution {
public int lengthOfLongestSubstring(String s) {

}
}
```

Python3:

```
class Solution:
def lengthOfLongestSubstring(self, s: str) -> int:
```