3. Longest Substring Without Repeating Characters

Medium 28686 1222 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.
```

Code

```python
class Solution:
    def lengthOfLongestSubstring(self, s: str) -> int:
        ans = 0
        letter = []
        for i in s:
            if i in letter:
                letter = letter[letter.index(i) + 1:]
            letter.append(i)
            ans = max(ans, len(letter))
        return ans

```

The purpose of the code is to receive a string and find the length of the longest substring which doesn't contain the same character more than once. The challenging part was to create an algorithm which can effectively check all different cases of substrings so that it can find the longest among them.