0003. 无重复字符的最长子串

ITCharge 大约 2 分钟

- 标签：哈希表、字符串、滑动窗口
- 难度：中等

题目链接

- 0003. 无重复字符的最长子串 - 力扣

题目大意

描述：给定一个字符串 s。

要求：找出其中不含有重复字符的最长子串的长度。

说明：

- $0 \leq s.length \leq 5 * 10^4$。
- s 由英文字母、数字、符号和空格组成。

示例：

- 示例 1：

```py
输入：s = "abcabcbb"
输出：3
解释：因为无重复字符的最长子串是 "abc"，所以其长度为 3。
```

- 示例 2：

```py
输入：s = "bbbbb"
输出：1
解释：因为无重复字符的最长子串是 "b"，所以其长度为 1。
```

解题思路

思路 1：滑动窗口（不定长度）

用滑动窗口 $window$ 来记录不重复的字符个数，$window$ 为哈希表类型。

1. 设定两个指针：$left$、$right$，分别指向滑动窗口的左右边界，保证窗口中没有重复字符。
2. 一开始，$left$、$right$ 都指向 0。
3. 向右移动 $right$，将最右侧字符 $s[right]$ 加入当前窗口 $window$ 中，记录该字符个数。
4. 如果该窗口中该字符的个数多于 1 个，即 $window[s[right]] > 1$，则不断右移 $left$，缩小滑动窗口长度，并更新窗口中对应字符的个数，直到 $window[s[right]] \leq 1$。
5. 维护更新无重复字符的最长子串长度。然后继续右移 $right$，直到 $right \geq len(nums)$ 结束。
6. 输出无重复字符的最长子串长度。

思路 1：代码

```py
class Solution:
    def lengthOfLongestSubstring(self, s: str) -> int:
        left = 0
        right = 0
        window = dict()
        ans = 0

        while right < len(s):
            if s[right] not in window:
                window[s[right]] = 1
            else:
                window[s[right]] += 1

            while window[s[right]] > 1:
                window[s[left]] -= 1
                left += 1
```

```
            ans = max(ans, right - left + 1)
            right += 1

        return ans
```

思路 1：复杂度分析

- **时间复杂度**：$O(n)$。
- **空间复杂度**：$O(|\sum|)$。其中 $\sum$ 表示字符集，$|\sum|$ 表示字符集的大小。