0151. 反转字符串中的单词

ITCharge 大约 2 分钟

- 标签：双指针、字符串
- 难度：中等

题目链接

- 0151. 反转字符串中的单词 - 力扣

题目大意

描述：给定一个字符串 `s` 。

要求：反转字符串中所有单词的顺序。

说明：

- **单词**：由非空格字符组成的字符串　　中使用至少一个空格将字符串中的单词分隔开。
- 输入字符串 `s` 中可能会存在前导空格、尾随空格或者单词间的多个空格。
- 返回的结果字符串中，单词间应当仅用单个空格分隔，且不包含任何额外的空格。
- $1 \leq s.length \leq 10^4$。
- `s` 包含英文大小写字母、数字和空格 `' '`
- `s` 中至少存在一个单词。

示例：

- 示例 1：

```py
输入：s = "  hello world  "
输出："world hello"
解释：反转后的字符串中不能存在前导空格和尾随空格。
```

- 示例 2：

```py
输入：s = "a good   example"
输出："example good a"
解释：如果两个单词间有多余的空格，反转后的字符串需要将单词间的空格减少到仅有一个。
```

解题思路

思路 1：调用库函数

直接调用 Python 的库函数，对字符串进行切片，翻转，然后拼合成字符串。

思路 1：代码

```py
class Solution:
    def reverseWords(self, s: str) -> str:
        return " ".join(reversed(s.split()))
```

思路 1：复杂度分析

- **时间复杂度**：$O(n)$，其中 n 是字符串 `s` 的长度。
- **空间复杂度**：$O(1)$。

思路 2：模拟

第二种思路根据 API 的思路写出模拟代码，具体步骤如下：

- 使用数组 `words` 存放单词，使用字符串变量 `cur` 存放当前单词。
- 遍历字符串，对于当前字符 `ch` 。
- 如果遇到空格，则：
  - 如果当前单词不为空，则将当前单词存入数组 `words` 中，并将当前单词置为空串
- 如果遇到字符，则：
  - 将其存入当前单词中，即 `cur += ch` 。
- 如果遍历完，当前单词不为空，则将当前单词存入数组 `words` 中。
- 然后对数组 `words` 进行翻转操作，令 `words[i]` , `words[len(words) - 1 - i]` 交换元素。
- 最后将 `words` 中的单词连接起来，中间拼接上空格，将其作为答案返回。

思路 2：代码

```py
class Solution:
    def reverseWords(self, s: str) -> str:
        words = []
        cur = ""
        for ch in s:
            if ch == ' ':
                if cur:
                    words.append(cur)
                    cur = ""
            else:
                cur += ch

        if cur:
            words.append(cur)

        for i in range(len(words) // 2):
            words[i], words[len(words) - 1 - i] = words[len(words) - 1 - i],
words[i]

        return " ".join(words)
```

思路 2：复杂度分析

- **时间复杂度**：$O(n)$，其中 n 是字符串 `s` 的长度。
- **空间复杂度**：$O(1)$。