【贪心】-环中最长子串

题目描述与示例

题目描述

给你一个字符串 `s` ,首尾相连成一个环形,请你在环中找出 `o` 字符出现了偶数次最长子字符串的长度。

输入描述

输入由一个小写字母组成的字符串 `s`

`1 <= s.lenth <= 5x10^5`

输出描述

输出是一个整数

示例一

输入

```
alolobo
```

输出

```
6
```

说明

最长子字符串之一是 `alolob` ,它包含 `2` 个 `o`

示例二

输入

```
looxdolx
```

输出

```
7
```

说明

最长子字符串 `oxdolxl` ,由于是首尾连接一起的,
所以最后一个 `x` 和开头的 `l` 是连接在一起的此字符串包含 `2` 个 `o`

示例三

输入

```
bcbcbc
```

输出

```
6
```

说明

这个示例中,字符串 `bcbcbc` 本身就是最长的,
因为 `o` 都出现了 `0` 次

解题思路

这道题属于带点脑筋急转弯性质的贪心题。乍一看可能会被**环型**这个条件干扰，把问题想复杂了。

实际上从题目给的几个例子容易分析，如果

- 字符 `o` 在**原字符串**中出现的次数为**偶数次**，为了使得所选子串尽可能长，那么把整个字符串都选上，一定是最长的符合要求的子串。此时子串长度为 `len(s)` 。
- 字符 `o` 在**原字符串**中出现的次数为**奇数次**，为了使得所选子串尽可能长，那么保留一个 `o` 不去选择，那么剩余字符串中 `o` 出现的次数必然为偶数次，把删除一个 `o` 之后的所有字符都选上，一定是最长的符合要求的子串。此时子串长度为 `len(s)-1` 。

因此只需要统计字符 `o` 出现的次数的奇偶性即可。

代码

Python

```python
# 题目：【贪心】2024E-环中最长子串
# 分值：100
# 作者：闭着眼睛学数理化
# 算法：贪心
# 代码看不懂的地方，请直接在群上提问


s = input()
num_o = 0
# 遍历s中的字符ch，统计字符"o"出现的次数
for ch in s:
    if ch == "o":
        num_o += 1

n = len(s)
# 如果num_o为偶数，输出n
# 如果num_o为奇数，输出n-1
print(n) if num_o % 2 == 0 else print(n-1)
```

Java

```java
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String s = scanner.nextLine();
        int num_o = 0;

        for (int i = 0; i < s.length(); i++) {
            if (s.charAt(i) == 'o') {
                num_o++;
            }
```

```
        }

        int n = s.length();
        if (num_o % 2 == 0) {
            System.out.println(n);
        } else {
            System.out.println(n - 1);
        }
    }
}

```

C++

```cpp
#include <iostream>
#include <string>
using namespace std;

int main() {
    string s;
    cin >> s;
    int num_o = 0;

    for (char ch : s) {
        if (ch == 'o') {
            num_o++;
        }
    }

    int n = s.length();
    if (num_o % 2 == 0) {
        cout << n << endl;
    } else {
        cout << n - 1 << endl;
    }

    return 0;
}

```

时空复杂度

时间复杂度： `O(N)` 。统计 `o` 出现的次数，需要一次遍历字符串 `s` 。

空间复杂度： `O(1)` 。仅需若干常数变量。