

删除回文子序列

难度简单 160

给你一个字符串 s ，它仅由字母 'a' 和 'b' 组成。每一次删除操作都可以从 s 中删除一个回文子序列。

返回删除给定字符串中所有字符（字符串为空）的最小删除次数。

「子序列」定义：如果一个字符串可以通过删除原字符串某些字符而不改变原字符顺序得到，那么这个字符串就是原字符串的一个子序列。

「回文」定义：如果一个字符串向后和向前读是一致的，那么这个字符串就是一个回文。

示例 1:

输入: $s = \text{"ababa"}$

输出: 1

解释: 字符串本身就是回文序列，只需要删除一次。

示例 2:

输入: $s = \text{"abb"}$

输出: 2

解释: $\text{"abb"} \rightarrow \text{"bb"} \rightarrow \text{""}.$

先删除回文子序列 "a"，然后再删除 "bb"。

示例 3:

输入: $s = \text{"baabb"}$

输出: 2

解释: $\text{"baabb"} \rightarrow \text{"b"} \rightarrow \text{""}.$

先删除回文子序列 "baab"，然后再删除 "b"。

```
class Solution:
    def removePalindromeSub(self, s: str) -> int:
        return 1 if s==s[::-1] else 2
        ...
```

由于字符串本身只含有字母 'a' 和 'b' 两种字符，题目要求每次删除回文子序列（不一定连续）而使得字符串最终为空。题目中只包含两种不同的字符，由于相同的字符组成的子序列一定是回文子序列，因此最多只需要删除 2 次即可删除所有的字符。删除判断如下：

如果该字符串本身为回文串，此时只需删除 1 次即可，删除次数为 1。

如果该字符串本身不是回文串，此时只需删除 2 次即可，比如可以先删除所有的‘a’，再删除所有的‘b’，删除次数为 2。

...