521 Longest Uncommon Subsequence I

2018年4月13日 19:46

Given a group of two strings, you need to find the longest uncommon subsequence of this group of two strings. The longest uncommon subsequence is defined as the longest subsequence of one of these strings and this subsequence should not be **any** subsequence of the other strings.

A **subsequence** is a sequence that can be derived from one sequence by deleting some characters without changing the order of the remaining elements. Trivially, any string is a subsequence of itself and an empty string is a subsequence of any string.

The input will be two strings, and the output needs to be the length of the longest uncommon subsequence. If the longest uncommon subsequence doesn't exist, return -1.

Example 1:

Input: "aba", "cdc"

Output: 3

Explanation: The longest uncommon subsequence is "aba" (or "cdc"),
because "aba" is a subsequence of "aba",
but not a subsequence of any other strings in the group of two strings.

Note:

1. Both strings' lengths will not exceed 100.
2. Only letters from a ~ z will appear in input strings.

来自 <https://leetcode.com/problems/longest-uncommon-subsequence-i/description/>

给定两个字符串，你需要从这两个字符串中找出最长的特殊序列。最长特殊序列定义如下：该序列为某字符串独有的最长子序列（即不能是其他字符串的子序列）。

子序列可以通过删去字符串中的某些字符实现，但不能改变剩余字符的相对顺序。空序列为所有字符串的子序列，任何字符串为其自身的子序列。

输入为两个字符串，输出最长特殊序列的长度。如果不存在，则返回 -1。

示例 :

输入: "aba", "cdc"

输出: 3

解析: 最长特殊序列可为 "aba" (或 "cdc")

说明:

1. 两个字符串长度均小于100。
2. 字符串中的字符仅含有 'a'~'z'。

Solution for Python3:

```
class Solution:
    def findLUSlength(self, a, b):
        """
        :type a: str
        :type b: str
        :rtype: int
        """
        return -1 if a == b else max(len(a), len(b))
```

Solution for C++:

```
class Solution {
public:
    int findLUSlength(string a, string b) {
        if (a == b)
            return -1;
        return a.length() > b.length() ? a.length() : b.length();
    }
};
```