

### 题目描述

给定字符串 target 和 source，判断 target 是否为 source 的子序列。

你可以认为target和 source 中仅包含英文小写字母。

字符串 source 可能会很长 (长度 $\sim$ 500,000), 而 target是个短字符串 (长度 $\leq$ 100)。

字符串的一个子序列是原始字符串删除一些（也可以不删除）字符而不改变剩余字符相对位置形成的新字符串。

(例如, "abc"是"aebxcd"的一个子序列, 而"ayb"不是)。

请找出最后一个子序列的起始位置。

### 输入描述

第一行为target, 短字符串 (长度  $\leq 100$ )

第二行为source, 长字符串 (长度  $\sim 500,000$ )

### 输出描述

最后一个子序列的起始位置，即最后一个子序列首字母的下标

## 备注

若在source中找不到target, 则输出-1。

## 用例

输入	abc abcaybec
输出	3
说明	这里有两个abc的子序列满足，取下标较大的，故返回3。

### 题目解析

此题是1

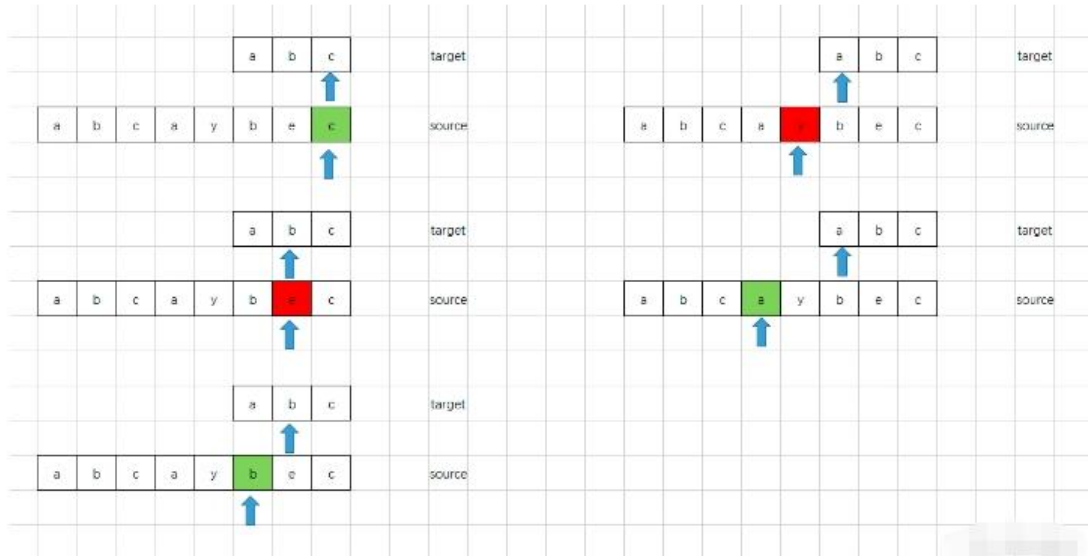
的进阶题。

此题不再好用正则来投机取巧了，因为source串中可能存在多个target子串，此时使用/a.\*b.\*c./正则就无法正确匹配了，因为此正则会直接匹配出abcaybec整个字符串，而不是某个子串。而想要构造一个只匹配子串，而不匹配整串的正则非常难，不适合机试。

因此，本题我们还可以使用指针来做。

创建一个指针cursor，来指向target的尾部target.length-1位置，而不是指向target的0位置

因为题目要求最后一个子序列首字母的下标，因此我们反向遍历target串，以及source串，这样的话，匹配到的第一个序列就是符合要求的。



反向遍历source字符串的每一个字符，与当前cursor指针指向的target串的字符比较，若相同，则cursor--，若不同，则cursor不变。

当cursor指向0位置时，此时若有source[i]的字符与target[cursor]相同，则i就是题解。

## Java算法源码

```
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.println(getResult(sc.nextLine(), sc.nextLine()));
8     }
9
10    public static int getResult(String target, String source) {
11        int cursor = target.length() - 1;
12        for (int i = source.length() - 1; i >= 0; i--) {
13            if (source.charAt(i) == target.charAt(cursor)) {
14                cursor--;
15                if (cursor < 0) return i;
16            }
17        }
18
19        return -1;
20    }
21 }
```

## JS算法源码

```
1  /* JavaScript Node ACM模式 控制台输入获取 */
2  const readline = require("readline");
3
4  const rl = readline.createInterface({
5    input: process.stdin,
6    output: process.stdout,
7  });
8
9  const lines = [];
10 rl.on("line", (line) => {
11   lines.push(line);
12
13   if (lines.length === 2) {
14     let target = lines[0];
15     let source = lines[1];
16     console.log(getValidSubStrIndex(target, source));
17
18     lines.length = 0;
19   }
20 });
21
22 function getValidSubStrIndex(target, source) {
23   let cursor = target.length - 1;
24   for (let i = source.length - 1; i >= 0; i--) {
25     if (source[i] === target[cursor]) {
26       cursor--;
27       if (cursor < 0) {
28         return i;
29       }
30     }
31   }
32   return -1;
33 }
```

## Python算法源码

```
1 # 输入数据
2 target = input()
3 source = input()
4
5
6 # 算法入口
7 def getResult():
8     cursor = len(target) - 1
9     for i in range(len(source)-1, -1, -1):
10         if source[i] == target[cursor]:
11             cursor -= 1
12             if cursor < 0:
13                 return i
14
15     return -1
16
17
18 # 调用算法
19 print(getResult())
```