

题目描述

给定a-z，26个英文字母小写字符串组成的字符串A和B，其中A可能存在重复字母，B不会存在重复字母，现从字符串A中按规则挑选一些字母可以组成字符串B。

挑选规则如下：

- 同一个位置的字母只能挑选一次，
- 被挑选字母的相对先后顺序不能被改变，
- 求最多可以同时从A中挑选多少组能组成B的字符串。

输入描述

输入为2行，第一行输入字符串a,第二行输入字符串b，行首行尾没有多余空格

输出描述

输出一行，包含一个数字，表示最多可以同时从a中挑选多少组能组成b的字符串，行末没有多余空格

用例

输入	badc bac
输出	1
说明	无

题目解析

本题求解可以参考

[LeetCode - 1419 数青蛙_伏城之外的博客-CSDN博客](#)

[华为机试 - 数大雁_伏城之外的博客-CSDN博客](#)

题目的用例不能说明问题，我们可以通过下面用例

bbadcbacdaccbbac
bac

a字符串

b	b	a	d	c	b	a	c	d	a	c	c	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

b字符串

b	0
a	0
c	0

b b | a | d | c | b | a | c | d | a | c | c | c | c | b | a | c |

↑

b 1 || a | 0 |
| c | 0 |

b b | a | d | c | b | a | c | d | a | c | c | c | c | b | a | c |

↑

b 2 || a | 0 |
| c | 0 |

伏城之外

已关注

3

3

2

专栏目录

已订阅

题目解析

本题求解可以参考

[LeetCode - 1419 数青蛙_伏城之外的博客-CSDN博客](#)

[华为机试 - 数大雁_伏城之外的博客-CSDN博客](#)

题目的用例不能说明问题，我们可以通过下面用例

bbadcbacdaccbbac
bac

a字符串

b	b	a	d	c	b	a	c	d	a	c	c	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

b字符串

b	0
a	0
c	0

b	b	a	d	c	b	a	c	d	a	c	c	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑

b	1
a	0
c	0

b	b	a	d	c	b	a	c	d	a	c	c	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑

b	2
a	0
c	0

b	b	a	d	c	b	a	c	d	a	c	c	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑

b	2
a	1
c	0

b	b	a	d	c	b	a	c	d	a	c	c	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑

b	2
a	1
c	0

b	b	a	d	c	b	a	c	d	a	c	c	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑

b	2
a	1
c	1

b	b	a	d	c	b	a	c	d	a	c	c	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑

b	3
a	1
c	1

b	b	a	d	c	b	a	c	d	a	c	c	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑

b	3
a	2
c	1

b	b	a	d	c	b	a	c	d	a	c	c	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑

b	3
a	2
c	2

b	b	a	d	c	b	a	c	d	a	c	c	c	c	b	a	c
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑

b	3
a	2
c	3

伏城之外

已关注

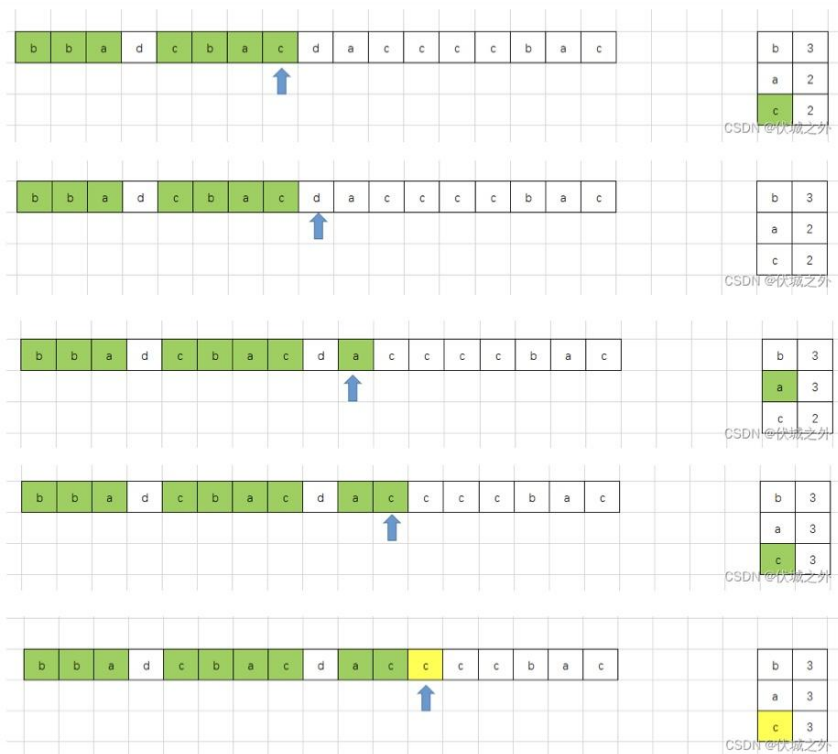
3

3

2

专栏目录

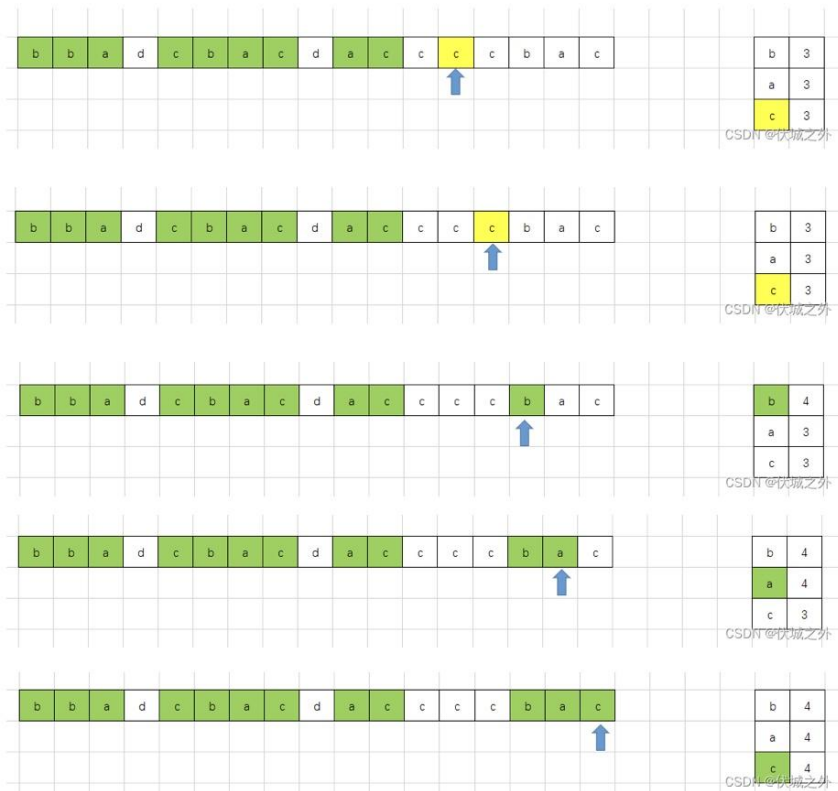
已订阅



注意：统计时，a的数量应该小于等于b的数量，c的数量应该小于等于a的数量，这样才能满足顺序要求：

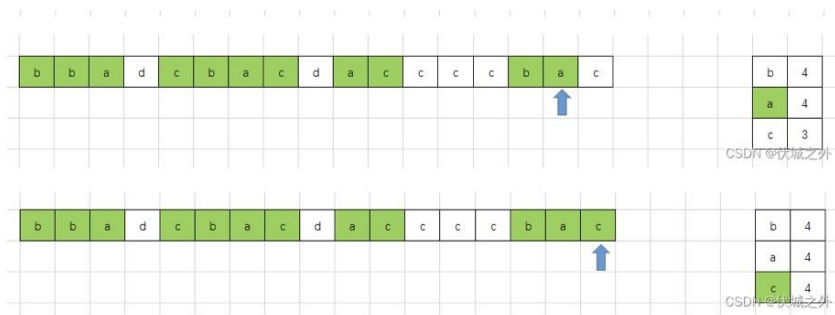
被挑选字母的相对先后顺序不能被改变

因此上面这步统计到的c不应该被计入。



由于 $c \leq b \leq a$ ，因此有几个c，字符串a中就能挑选出几个字符串b。

上面算法只需要一次遍历，即可完成题解，时间复杂度 $O(n)$



由于 $c \leq b \leq a$, 因此有几个c, 字符串a中就能挑选出几个字符串b。

上面算法只需要一次遍历, 即可完成题解, 时间复杂度 $O(n)$

JavaScript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   console.log(getResult(lines[0], lines[1]));
15   lines.length = 0;
16 }
17 });
18
19 function getResult(a, b) {
20   // idxs 对象记录字符串b中每个字符的索引
21   const idxs = {};
22   for (let i = 0; i < b.length; i++) {
23     idxs[b[i]] = i;
24   }
25
26   // count 对象用于记录遍历字符串a 每个字符串过程中, 统计到的符合顺序要求的字符串b 中字符出现次数
27   const count = new Array(b.length).fill(0);
28
29   for (let c of a) {
30     const idx = idxs[c];
31     // 下面判断逻辑请看图解
32     if (idx !== undefined && (idx === 0 || count[idx] < count[idx - 1])) {
33       count[idx]++;
34     }
35   }
36
37   return count.at(-1);
38 }
```

Java算法源码

```
1  import java.util.HashMap;
2  import java.util.Scanner;
3
4  public class Main {
5    public static void main(String[] args) {
6      Scanner sc = new Scanner(System.in);
7
8      String a = sc.next();
9      String b = sc.next();
10
11      System.out.println(getResult(a, b));
12    }
13
14    public static int getResult(String a, String b) {
15      // idxs 对象记录字符串b中每个字符的索引
16      HashMap<Character, Integer> idxs = new HashMap<>();
17      for (int i = 0; i < b.length(); i++) {
18        Character c = b.charAt(i);
19        idxs.put(c, i); // B不会存在重复字母
20      }
21
22      // count 对象用于记录遍历字符串a 每个字符串过程中, 统计到的符合顺序要求的字符串b 中字符出现次数
23      int[] count = new int[b.length()];
24      for (int i = 0; i < a.length(); i++) {
25        Character c = a.charAt(i);
26
27        if (idxs.containsKey(c)) {
28          int idx = idxs.get(c);
29          // 下面判断逻辑请看图解
```

伏城之外 已关注

3 2

专栏目录

已订阅

Java算法源码

```
1 import java.util.HashMap;
2 import java.util.Scanner;
3
4 public class Main {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         String a = sc.next();
9         String b = sc.next();
10
11         System.out.println(getResult(a, b));
12     }
13
14     public static int getResult(String a, String b) {
15         // idxs 对象记录字符串b中每个字符的索引
16         HashMap<Character, Integer> idxs = new HashMap<>();
17         for (int i = 0; i < b.length(); i++) {
18             Character c = b.charAt(i);
19             idxs.put(c, i); // B 不会存在重复字母
20         }
21
22         // count 对象用于记录遍历字符串a每个字符串过程中，统计到的符合顺序要求的字符串b中字符出现次数
23         int[] count = new int[b.length()];
24         for (int i = 0; i < a.length(); i++) {
25             Character c = a.charAt(i);
26
27             if (idxs.containsKey(c)) {
28                 int idx = idxs.get(c);
29                 // 下面判断逻辑请看图解
30                 if (idx == 0 || count[idx] < count[idx - 1]) {
31                     count[idx]++;
32                 }
33             }
34         }
35
36         return count[count.length - 1];
37     }
38 }
```

Python算法源码

```
1 # 输入获取
2 a = input()
3 b = input()
4
5
6 # 算法入口
7 def getResult(a, b):
8     # idxs 对象记录字符串b中每个字符的索引
9     idxs = {}
10     for i in range(len(b)):
11         idxs[b[i]] = i
12
13     # count 对象用于记录遍历字符串a每个字符串过程中，统计到的符合顺序要求的字符串b中字符出现次数
14     count = [0] * len(b)
15     for c in a:
16         idx = idxs.get(c)
17
18         # 下面判断逻辑请看图解
19         if idx is not None and (idx == 0 or count[idx] < count[idx - 1]):
20             count[idx] += 1
21
22     return count[-1]
23
24
25 # 算法调用
26 print(getResult(a, b))
```