

华为OD机试 - 最长公共后缀 (Java & JS & Python)

原创 伏城之外 已于 2023-07-03 01:43:42 修改 701 收藏 2

版权

分类专栏: 华为OD机试AB (Java & JS & Python) 文章标签: 算法 华为机试 Java JavaScript Python



题目描述

编写一个函数来查找字符串数组中的最长公共后缀；

如果不存在公共后缀，返回固定字符串：@Zero。

补充说明：

1. 字符串长度范围：[2, 1000]
2. 字符串中字符取值范围为[1, 126]

输入描述

无

输出描述

无

用例

输入	["abc", "bbc", "c"]
输出	"c"
说明	返回公共后缀: c

输入	["aa", "bb", "cc"]
输出	"@Zero"
说明	不存在公共后缀，返回固定结果: @Zero

题目解析

本题应该是采用核心代码模式，非ACM模式，因此不需要我们处理输入输出。

下面代码仍然以ACM模式实现，但是会将输入输出处理和核心代码分离。考试时，只需要写出核心代码即可。

关于核心代码实现，我的思路如下



假设输入的字符串数组为strs，则可以：

1. 将strs[0]假设为最长公共后缀suffix
2. 之后，再找出suffix和strs[1]的最长公共后缀，并覆盖给suffix，按此逻辑继续找出suffix和其他strs[i]的最长公共后缀

当然，在上面过程中，一旦发现suffix == ""，即最长公共后缀是空串，则可以直接返回@Zero。

否则，返回suffix。

```
1 import java.util.Arrays;
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 line = sc.nextLine();
9         String[] strings =
10             Arrays.stream(line.substring(1, line.length() - 1).split(","))
11                 .map(s -> s.substring(1, s.length() - 1))
12                 .toArray(String[]::new);
13
14         System.out.println(getResult(strings));
15     }
16
17     public static String getResult(String[] strings) {
18         // 假设第0个字符串就是最长公共后缀
19         String suffix = strings[0];
20
21         // 从第1个字符串开始，求解和最长公共后缀suffix的最长公共后置
22         for (int i = 1; i < strings.length; i++) {
23             suffix = getLCS(suffix, strings[i]);
24
25             // 如果最长公共后缀为“”，则直接返回"@Zero"
26             if ("".equals(suffix)) return "@Zero";
27         }
28
29         return suffix;
30     }
31
32     /**
33      * 求两个字符串的最长公共后缀
34      *
35      * @param s1 字符串1
36      * @param s2 字符串2
37      * @return 两个字符串的最长公共后缀
38      */
39     public static String getLCS(String s1, String s2) {
```

```
40 // 如果尾字符不同，则没有公共后缀
41 if (charAt(s1, -1) != charAt(s2, -1)) return "";
42
43 // 最长公共后缀的长度上限是：两个字符串中较短的那个的长度值
44 int maxLen = Math.min(s1.length(), s2.length());
45
46 // 开始逐位比较
47 for (int i = -2; i >= -maxLen; i--) {
48     // 如果某位对应字符不同，则该位后面的就是最长公共后缀
49     if (charAt(s1, i) != charAt(s2, i)) return s1.substring(s1.length() + i + 1);
50 }
51
52 // 如果比较完了，都没有发现对应位不同字符，则说明，两个字符串中较短者本身就是最长公共后缀
53 return s1.substring(s1.length() - maxLen);
54 }
55
56 // 负数索引找字符，比如索引-1 等价于 索引s.length-1
57 public static char charAt(String s, int negativeIndex) {
58     return s.charAt(s.length() + negativeIndex);
59 }
60 }
```