

题目描述

输入一个由n个大小写字母组成的字符串

按照ASCII码值从小到大进行排序

输出该字符串所在字符串中的位置索引(字符串的第一个位置索引为0)

k应大于字符串长度则输出最大ASCII码值的字母所在字符串的位置索引

如果有重复字母则输出字母的最小位置索引

输入描述

第一行输入一个由大小写字母组成的字符串

第二行输入k，k应大于0，k可以大于输入字符串的长度

输出描述

输出字符串中第k个最小ASCII码值的字母所在字符串的位置索引

k应大于字符串长度则输出最大ASCII码值的字母所在字符串的位置索引

如果有多个最小ASCII码值的字母存在重复 则输出该字母的最小位置索引

用例

输入	ABCSMFG
输出	5
说明	根据ASCII码值排序，第三个ASCII码值的字母为F F在字符串中位置索引为5(0为字符串的第一个字母位置索引)
输入	PMSCARHMQ
输出	6
说明	根据ASCII码值排序前四个字母为AABH由于B重复所以取B 的第一个最小位置索引6 而不是第二个B的位置索引8

题目解析

简单的 字符串操作 题。

2023.05.20 补充了第二个用例

根据第二个用例来看，题目要找到第k个，不是去重+升序后的第k个，而是排序后的第k个。

JavaScript算法源码

```
1 // 题目描述: 输入一个由n个大小写字母组成的字符串
2 // 按照ASCII码值从小到大进行排序
3 // 输出该字符串所在字符串中的位置索引(字符串的第一个位置索引为0)
4 // k应大于字符串长度则输出最大ASCII码值的字母所在字符串的位置索引
5 // 如果有重复字母则输出字母的最小位置索引
6
7 const readline = require('readline');
8
9 const rl = readline.createInterface({
10   input: process.stdin,
11   output: process.stdout,
12 });
13
14 const lines = [];
15 rl.on('line', (line) => {
16   lines.push(line);
17 });
18
19 if (lines.length === 2) {
20   const [str, k] = lines;
21   console.log(getIndex(str, k));
22 }
23
24 lines.length = 0;
25
26 function getIndex(str, k) {
27   if (k > str.length) k = str.length;
28
29   const tar = [...str].sort();
30   return str.indexOf(tar[k - 1]);
31 }
```

Java算法源码

```
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 str = sc.next();
9         int k = sc.nextInt();
10
11         System.out.println(getResult(str, k));
12     }
13
14     public static int getResult(String str, int k) {
15         char[] chars = str.toCharArray();
16         Arrays.sort(chars);
17
18         if (k > str.length()) k = str.length();
19
20         char tar = chars[k - 1];
21         return str.indexOf(tar);
22     }
23 }
```

Python算法源码

```
1 # 题目描述:
2 # 输入一个由n个大小写字母组成的字符串
3 # 按照ASCII码值从小到大进行排序
4 # 输出该字符串所在字符串中的位置索引(字符串的第一个位置索引为0)
5 # k应大于字符串长度则输出最大ASCII码值的字母所在字符串的位置索引
6 # 如果有重复字母则输出字母的最小位置索引
7
8 # 解题思路:
9 # 1. 读取输入
10 # 2. 将字符串转换为列表
11 # 3. 对列表进行排序
12 # 4. 找到第k个元素
13 # 5. 返回该元素在原始字符串中的索引
14
15 # 代码实现:
16 def getResult(s, k):
17     chars = list(s)
18     chars.sort()
19
20     if k > len(s):
21         k = len(s)
22
23     tar = chars[k - 1]
24     return s.index(tar)
25
26 # 测试用例:
27 print(getResult('ABCSMFG', 5))
```