```
● CSDN 博客 下载 学习 社区 GitCode 云服务 猿如意
                     版权
                     分类专栏: 华为OD机试2023 (JAVA) 文章标签: java 算法 华为

#500-Adeco
华为OD机试2023 (... 专栏收录该内容
¥59.90

#4.6 # 和
10 该专栏为热销专栏榜 第1名
¥99.00

                                                          750 订阅 214 篇文章 (已订阅
                                                                                    超级会员免费看
                   华为OD机试真题,2023年度机试题库全覆盖,刷题指南点<u>这里</u>
                   字符串重新排序
                   知识点排序数组
                   时间限制: 1s 空间限制: 256MB 限定语言: 不限
                   题目描述:
                   给定一个字符串s,s包含以空格分隔的若干个单词,请对s进行如下处理后输出:
                   1、单词内部调整:对每个单词字母重新按字典序排序;
                   2、单词间顺序调整:
                     1) 统计每个单词出现的次数,并按次数降序排列;
                     2) 次数相同时,按单词长度升序排列;
                     3) 次数和单词长度均相同时,按字典序升序排列。
                   请输出处理后的字符串,每个单词以一个空格分隔。
                   输入描述:
                   一行字符串,每个字符取值范围: [a-zA-Z0-9]以及空格,字符串长度范围: [1, 1000]
                   输出描述:
                   重新排序后的字符串,每个单词间隔1个空格,且首尾无空格
                   示例1
                   输入:
                   This is an apple
                   输出:
                   an is This aelpp
                   示例2
                   输入:
                   My sister is in the house not in the yard
                   输出:
                   in in eht eht My is not adry ehosu eirsst
                   解题思路:
                   1、使用map来记录单词的使用次数(根据题意需要先对单词进行字母排序,也就是说只要字母都相同的就是同一
                   个单词,不管其顺序)
                   2、将map转化为list进行sort排序,排序规则:
                    1. 使用次数进行降序(map中value大的排在前面)
                    2. 使用次数相同时,根据key的长度升序(key的长度短的排在前)
                    3. 长度相同时,根据字典序进行升序
                   3、最后按序输出list中的字符
                   代码:
                       public class Main{
                     1 |
                     2
                     3
                          public static void main(String[] args) {
                     4
                              Scanner sc = new Scanner(System.in);
                     5
                     6
                              String[] strings = sc.nextLine().split(" ");
                     7
                     8
                              Map<String,Integer> map = new HashMap<>();
                     9
                              for( int i=0; i<strings.length; i++){</pre>
                    10
                                 char[] chars = strings[i].toCharArray();
                    11
                                 Arrays.sort(chars);
                    12
                                                    //对单子中的字母进行字典排序
                                 String temp = String.valueOf(chars);
                    13
                                 map.put( temp, map.getOrDefault( temp, 0) + 1); //统计每个单词出现的次数
                    14
                    15
                    16
                    17
                              List<Map.Entry<String,Integer>> list = new ArrayList<>(map.entrySet());
                    18
                              list.sort(( a, b)->{
                    19
                                 if(b.getValue() < a.getValue()){</pre>
                    20
                                    return -1; //按次数降序
                    21
                                 }else if(b.getValue() == a.getValue()){
                    22
                                                                    //次数相同
                    23
                                    if(b.getKey().length() > a.getKey().length()){
                                       return -1; //按字母长度升序
                    24
                                    }else if(b.getKey().length() == a.getKey().length()){ //字母长度相同
                    25
                                       return a.getKey().compareTo(b.getKey());
                    26
                                                                           //按字典序升序
                    27
                                    }
                    28
                    29
                                 return 1;
                              });
                    30
                    31
                    32
                              String res = "";
                              for(int i=0; i<list.size(); i++){</pre>
                    33
                                 for(int j=0;j<list.get(i).getValue();j++){</pre>
                    34
                                    res += list.get(i).getKey() + " ";
                    35
                    36
                              }
                    37
                    38
                              System.out.println(res.substring(0, res.length()-1));
                    39
                    40
                    41 }
                   满分答案:
                       import java.util.*;
                     1 |
                     2
                       public class Main {
                     4
                          public static void main(String[] args) {
                     5
                     6
                     7
                              Scanner sc = new Scanner(System.in);
                     8
                     9
                              String[] strings = sc.nextLine().split(" ");
                              Map<String,Integer> map = new HashMap<>();
                    10
                    11
                              for(int i=0;i<strings.length;i++){</pre>
                    12
                                 String temp = chuli(strings[i]);
                    13
                                 map.put(temp,map.getOrDefault(temp,0)+1);
                    14
                    15
                              }
                    16
                              List<Map.Entry<String,Integer>> list = new ArrayList<>(map.entrySet());
                    17
                    18
                              list.sort((a,b)->{
                    19
                                 if(b.getValue()<a.getValue()){</pre>
                    20
                    21
                                    return -1;
                                 }else if(b.getValue() == a.getValue()){
                    22
                                    if(b.getKey().length() > a.getKey().length()){
                    23
                                       return -1;
                    24
                                    }else if(b.getKey().length() == a.getKey().length()){
                    25
                                       return a.getKey().hashCode()-b.getKey().hashCode();
                    26
                    27
                    28
                    29
                                 return 1;
                              });
                    30
                    31
                              String res = "";
                    32
                    33
                              for(int i=0;i<list.size();i++){</pre>
                                 for(int j=0;j<list.get(i).getValue();j++){</pre>
                    34
                    35
                                    res += list.get(i).getKey() + " ";
                    36
                    37
                    38
                    39
                    40
                              System.out.println(res.substring(0, res.length()-1));
                    41
                          }
                    42
                          public static String chuli(String s){
                    43
                    44
                              List<Character> list = new ArrayList<>();
                    45
                              for(int i=0;i<s.length();i++){</pre>
                    46
                                 list.add(s.charAt(i));
                    47
                    48
                              Collections.sort(list);
                    49
                              String res = "";
                    50
                              for(int i=0;i<list.size();i++){</pre>
                    51
                                 res += list.get(i);
                    52
                    53
                    54
                    55
                              return res;
                    56
                    57
                    58
                    59 }
                 跳 文章知识点与官方知识档案匹配,可进一步学习相关知识
                 算法技能树 > 首页 > 概览 40097 人正在系统学习中
                    若博豆(已关注)
                                                                                   专栏目录
                                                                                            已订阅
                华为OD机试 - 字符串重新排列、字符串重新排序(Java & JS & Python)
                                                                                 2022.Q4 已支持Java、Python、JavaScript,排序
                                                                                    wddpfx的博客 🧿 161
                华为机试真题 C++ 实现【字符串重新排列】【2022.11 Q4新题】
                一行<mark>字符串</mark>,每个字符取值范围:【a-zA-z0-9】以及空格,<mark>字符串</mark>长度范围:【1,1000】1、单词内部调整:对每个单词字母<mark>重新</mark>按字…
                评论 2条>
                                                                                            写评论
                    樊城太守 热评 这个题的答案不正确吗 两个测试用例都通不过
                 【华为OD机试真题 JAVA】字符统计及重排
                                                                                标题:字符统计及重排 给出一个仅包含字母的<mark>字符串</mark>,不包含空格,统计<mark>字符串</mark>中各个字母(区分大小写)出现的次数,并按照字母出...
                华为机试真题 Python 实现【字符串重新排列】【2022.11 Q4新题】
                                                                      给定一个<mark>字符串</mark>s,s包括以空格分隔的若干个单词,请对s进行如下处理后输出: 1、单词内部调整:对每个单词字母<mark>重新</mark>按字典序<mark>排序</mark> ....
                程序员外包, 你还在等什么?
                程序开发外包网
                 【华为机试真题 JAVA】字符统计及重排-100
                                                                              给出一个仅包含字母的<mark>字符串</mark>,不包含空格,统计<mark>字符串</mark>中各个字母(区分大小写)出现的次数,并按照字母出现次数从大到小的顺序....
                每日一题 找到Ascii码排序为k的索引
                                                                                 package leetcode; import java.util.Arrays; import java.util.HashMap; import java.util.Map; import java.util.Scanner; /** * 输入一个由n个...
                 【华为机试真题 Python实现】工号不够用了怎么办(员工工号问题)【2022 Q1 Q2 |100分】 不太灵光的程序员 ◎ 3619
                 《华为机试真题》专栏含牛客网华为专栏、华为面经试题、华为OD机试真题。如果您在准备华为的面试,期间有想了解的可以私信我,…
                字符串排序
                                                                        题目描述 编写一个程序,将输入<mark>字符串</mark>中的字符按如下规则<mark>排序</mark>。 规则 1: 英文字母从 A 到 Z 排列,不区分大小写。   如,输入: …
                每日一题@76最后一个单词
                                                                                 最后一个单词的长度 给你一个字符串 s,由若干单词组成,单词之间用空格隔开。返回字符串中最后一个单词的长度。如果不存在最后……
                 【满分】【华为OD机试真题2023 JAVA】字符串解密
                                                                                字符串解密 知识点数组字符串排序 时间限制:1s空间限制:256MB限定语言:不限 题目描述:给定两个字符串string1和string2。string...
                华为OD机试真题 C++ 实现【字符串重新排列】【2022.11 Q4新题| 100分】
                                                                                  华为OD机试真题 C++ 实现【字符串重新排列】【2022.11 Q4新题】
                题目0092-重组字符串
                                                                                  给定一个非空<mark>字符串</mark>S,其被N个'-'分隔成N+1的子串,给定正整数K,要求除第一个子串外,其余的子串每K个字符组成新的子串,并用'...
                java面试题及答案2022,java2022最新面试题及答案
                                                                                   发现网上很多Java面试题都没有答案,所以花了很长时间搜集整理出来了这套Java面试题大全,希望对大家有帮助哈~本套Java面试题...
                华为OD机试 - 字符串加密
                                                                                  字符串加密,动态规划
                算法题:给定一个字符串,字符串中包含一些空格,将字符串中由空格隔开的单词反序,并反… XreqcxoKiss的博客 ① 687
                对于一个长度为 n 字符串,首先这个字符串中包含着一些空格,就像"Hello World"一样,然后我们要做的是把这个字符串中由空格隔开...
                 【华为OD机试真题 python】字符串排序 【2022 Q4 | 100分】
                                                                                   无痕de泪的博客 ① 407
                华为OD机试 Python OD笔试 字符串排序
                                                                                                       Beta
                                                                                                      【算法练习】字符串重新排序 最新发布
                算法练习-华为OD
                                                                                                       Beta
                这段代码实现的<mark>算法</mark>是对一组单词进行<mark>排序,排序</mark>规则为首先按照单词出现频率降序<mark>排序</mark>,其次按照单词长度升序<mark>排序</mark>,最后按照单词……
                                                                                                      "相关推荐"对你有帮助么?
                                                                                                      6)
                                              非常没帮助
                                                                       举报
                       关于我们 招贤纳士 商务合作 寻求报道 ☎ 400-660-0108 ☑ kefu@csdn.net ⑤ 在线客服 工作时间 8:30-22:00
                      公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心
                       家长监护 网络110报警服务 中国互联网举报中心 Chrome商店下载 账号管理规范 版权与免责声明 版权申诉 出版物许可证 营业执照
```

©1999-2023北京创新乐知网络技术有限公司

① 发布