

【华为OD机试真题2023 JAVA】最大数字

原创

若博豆

已于 2023-02-08 15:02:36 修改

807 收藏 3

版权

分类专栏：


华为OD机试2023 (JAVA)

文章标签：

java

算法

华为

 华为OD机试2023 (...)

专栏收录该内容


¥59.90

¥99.00

750 订阅

216 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，刷题指南点这里

求最大数字

知识点 [单调栈](#)

时间限制：1s 空间限制：256MB 限定语言：不限

题目描述：

给定一个由纯数字组成以字符串表示的数值，现要求字符串中的每个数字最多只能出现2次，超过的需要进行删除；删除某个重复的数字后，其它数字相对位置保持不变。

如"34533"，数字3重复超过2次，需要删除其中一个3，删除第一个3后获得最大数值"4533"

请返回经过删除操作后的最大的数值，以字符串表示。

输入描述：

第一行为一个纯数字组成的字符串，长度范围：[1,100000]

输出描述：

输出经过删除操作后的最大的数值

示例1

输入：

34533

输出：

4533

示例2

输入：

5445795045

输出：

5479504

解题思路：

1. 根据题意，所有数字不超过2个，对输入的字符串进行去重处理，所有数字最多保留2个

2. 对步骤1处理完的数组进行全排列，同时需要判断排列后的字符串是否为原始字符串的子串，最后求出其中最大值

代码：

```
1 public class Main{
2
3     public static List<Integer> resList = new ArrayList<>();
4     public static String[] inputStrs;
5
6     public static void main(String[] args) {
7
8         Scanner sc = new Scanner(System.in);
9
10        inputStrs = sc.nextLine().split("");
11
12        Map<String, Integer> map = new HashMap<>();
13
14        for(int i=0; i<inputStrs.length; i++){ // 求出所有数字的出现次数
15            map.put( inputStrs[i], map.getOrDefault(inputStrs[i], 0) + 1);
16        }
17
18        List<String> list = new ArrayList<>();
19        for(Map.Entry<String,Integer> m : map.entrySet()){
20            String num = m.getKey();
21            list.add(num);
22            if(m.getValue() >= 2){ // 次数大于等于2只记两次
23                list.add(num);
24            }
25        }
26
27        String[] strNum = new String[list.size()];
28        list.toArray(strNum); // 集合转化为数组
29        combine( strNum,0, strNum.length-1);
30        Collections.sort(resList);
31
32        System.out.println(resList.get(resList.size()-1));
33
34    }
35
36    public static void swap(String[] strings, int indexa, int indexb){
37
38        String temp = strings[indexa];
39        strings[indexa] = strings[indexb];
40        strings[indexb] = temp;
41
42    }
43
44    /**
45     * 对存在的数字进行全排列，找出其中最大的值
46     * @param str 存在数字数组
47     * @param start 进行交换的index
48     * @param end 数组的最后一个index
49     */
50    public static void combine(String[] str, int start, int end){
51
52        if(start == end){
53            String numStr = "";
54            for (String s : str){
55                numStr += s; // 将数组中的数字进行拼接
56            }
57            if(!resList.contains(Integer.valueOf(numStr)) && isChild(strs)){ // 删除满足
58                resList.add(Integer.valueOf(numStr));
59            }
60        }else {
61            for(int i=start; i<strs.length; i++){
62                if(i != start && str[i].equals(strs[start])){ // 用来交换的索引不同且数字相等
63                    continue;
64                }
65                swap(strs, i, start);
66                if(strs[0].equals("0")){
67                    continue; // 首位为0不满足
68                }
69                combine(strs,start+1, end);
70                swap(strs, i, start);
71            }
72        }
73    }
74
75    /**
76     * 判断数组是否为输入数组的子串
77     * @param child
78     * @return
79     */
80    public static boolean isChild(String[] child){
81
82        int index = 0; // 父数组索引
83        boolean isOver = false; // 父数组是否遍历结束
84        int count = 0;
85        for(int i=0; i<child.length; i++){
86            String str = child[i];
87            for(int j = index; j<inputStrs.length; j++){
88                String temp = inputStrs[j];
89                if(j == inputStrs.length-1){
90                    isOver = true; // 父数组遍历结束了
91                }
92                if(temp.equals(str)){
93                    index = j + 1; // 存在该字符，则更新count
94                    count ++;
95                    break;
96                }
97            }
98            if(isOver){ // 父数组遍历完成且index没有刷新
99                break;
100            }
101        }
102        return count == child.length;
103    }
104 }
105 }
```

算法二：


```
1 public class Main{
2
3     public static void main(String[] args) {
4
5         Scanner sc = new Scanner(System.in);
6         String str = sc.nextLine();
7
8         /**
9          * key: 数字
10         * value: 数字出现的次数
11         */
12        Map<Integer, List<Integer>> map = new HashMap<>();
13        int[] nums = new int[str.length()]; // 转化为数组形式
14        for (int i = 0; i < str.length(); i++) {
15            List<Integer> tempList = new ArrayList<>();
16            int num = str.charAt(i) - '0';
17            nums[i] = num;
18            if(map.containsKey(num)){
19                tempList = map.get(num);
20                tempList.add(i);
21            }else {
22                tempList.add(i);
23                map.put( num, tempList);
24            }
25        }
26
27        List<Map.Entry<Integer, List<Integer>>> list = new ArrayList<>();
28        for(Map.Entry<Integer, List<Integer>> m : map.entrySet()){
29            if(m.getValue().size() >2){ // 将出现超过2次的数字放在一个集合中
30                list.add(m);
31            }
32        }
33
34        list.sort((a,b) -> {
35            return a.getKey() - b.getKey();
36        });
37
38        for(Map.Entry<Integer, List<Integer>> m : list){
39            handle( m.getValue(), nums);
40        }
41
42        String res = "";
43        for(int i : nums){
44            if(i != -1){
45                res += i;
46            }
47        }
48        System.out.println(res);
49    }
50
51    public static void handle( List<Integer> list, int[] nums){
52
53        while (list.size() > 2){
54            int index1 = list.get(0);
55            if(searchNum( nums, index1 + 1) > nums[index1]){
56                list.remove(0);
57                nums[index1] = -1;
58                continue;
59            }
60            int index2 = list.get(1);
61            if(searchNum( nums, index2 + 1) > nums[index2]){
62                list.remove(1);
63                nums[index2] = -1;
64                continue;
65            }
66            int index3 = list.get(2);
67            list.remove(2);
68            nums[index3] = -1;
69        }
70    }
71
72    public static int searchNum(int[] nums, int index){
73
74        int res = 0;
75        for(int i=index; i<nums.length; i++){
76            if(nums[i] != -1){
77                res = nums[i];
78                break;
79            }
80        }
81        return res;
82    }
83
84    }
85 }
86 }
```

📖 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

华为OD机试 - 最大数字 (Java & JS & Python) qfc_128220的博客 1267
2022.Q4 已支持Java、JavaScript、Python，考察逻辑分析

华为OD机试真题.pdf 04-15
华为机试真题（非牛客网刷题题）OD考试真题，不定期更新，文档含代码解答

评论 8条 >

 樊城太守 热评

public class test10_02 { public static void main(String[] args) { Scanner sc = new Scanner(System.in); String s = sc.nextLine(); HashMap<Character, Integer> hashMap = new HashMap<>(); char[] chars = s.toCharArray(); StringBuilder stringBuilder = new StringBuilder(); for (int i = chars.length - 1; i >= 0; i--) { char c = chars[i]; if (hashMap.containsKey(c) && hashMap.get(c) == 2) { char charAt = stringBuilder.charAt(stringBuilder.length() - 1); if (stringBuilder.length() > 0 && Integer.valueOf(charAt) < Integer.valueOf(c)){ stringBuilder.append(c); int index = stringBuilder.indexOf(String.valueOf(c)); stringBuilder.deleteCharAt(index); } continue; } stringBuilder.append(c); if (hashMap.containsKey(c)) { hashMap.put(c, hashMap.get(c) + 1);

外企德科OD真题 带超链接 06-05
外企德科OD真题 带超链接

【华为OD机试真题 python】最大数字 【2023 Q1 | 200分】 无限de泪痕的博客 437
华为OD机试 Python OD笔试题 最大数字 给定一个由纯数字组成以字符串表示的数值，现要求字符串中的每个数字最多只能出现2次，超过...

差旅管理系统
差旅报销管理系统

【华为OD机试真题 Java】字符串删除后的最大值 qq_40642440的博客 113
给定一个由纯数字组成以字符串表示的数值，现要求字符串中的每个数字最多只能出现2次，超过的需要进行删除；删除某个重复的数字...

【华为机试真题 JAVA】字符串删除后的最大数值-100 weixin_528940479的博客 858
给定一个由纯数字组成以字符串表示的数值，现要求字符串中的每个数字最多只能出现2次，超过的需要进行删除；删除某个重复的数字...

【华为OD机试 c++】最大数字 【2023 Q1考试题 A卷】 最新发布 goldamour的博客 59
【华为OD机试 c++】最大数字 【2023 Q1考试题 A卷】200分

华为OD机试 - 组成最大数 (Java) | 机试刷题思路 【2023】 梦想橡皮擦，专栏100例写作模式先行者，现象级专... 1240
组成最大数 小组中每位都有一张卡片，卡片上是 6 位内的正整数， 将卡片连起来可以组成多种数字，计算组成的最大数字。输入输出描...

找工作碎碎念-外企德科面试经历-与工作未来发展（深圳华为OD面经） 宇华的博文 5072
外企德科,华为OD面经

【华为OD机试真题 python】二元组个数 【2022 Q4 | 100分】 无限de泪痕的博客 421
华为OD机试 Python OD笔试题 二元组个数：给定两个数组a，b，若a[i] == b[j] 则称 [i, j] 为一个二元组，求在给定两个数组中，二元组的...

【华为OD机试真题 Python】字符串删除后的最大值 A_D_I_O_A_S的博客 488
给定一个由纯数字组成以字符串表示的数值，现要求字符串中的每个数字最多只能出现2次，超过的需要进行删除；删除某个重复的数字...

某厂机试算法刷题一览 唐门推荐 @_南先生的博客 1万+
目录 1. 【寻找身高相近的小朋友】 2. 【火星文计算】 3. 【最长的指定瑕能量的元音字符串】 4. 【消消乐游戏】 5. 【判断字符串子序列】 6...

【华为OD机试真题 C++】计算最大乘积 【2022 Q4 | 100分】 jidanza66689的博客 102
华为OD机试 C++ OD笔试题 计算最大乘积：给定一个元素类型为小写字母的数组，请计算两个没有相同字符的元素长度乘积的最大值，...

【华为OD机试真题 JAVA】组成最大数 【2022 Q4 | 100分】 pengzongy的博客 638
华为OD机试 Java OD笔试题 组成最大数：小组中每位都有一张卡片，卡片上是6位内的正整数，将卡片连起来可以组成多种数字，计算组...

【华为OD机试真题 C++】考勤信息 【2022 Q4 | 100分】 jidanza66689的博客 134
华为OD机试 C++ OD笔试题 考勤信息：公司用一个字符串来表示员工的出勤信息，需根据员工出勤信息，判断本次是否获得出勤奖

【华为OD机试真题 C++】数组拼接 【2022 Q4 | 100分】 jidanza66689的博客 86
华为OD机试 C++ OD笔试题 数组拼接：现在有多组整数数组，需要将它们合并成一个新的数组。合并规则，从每个数组里按顺序取出固定...

【华为OD机试真题 python】迷宫问题 weixin_42530333的博客 72
考试题目大同小异，练习真题是通过考试的捷径。专栏提供题目详细描述、解题思路和完整代码，不保证通过哦。

【华为OD机试真题 C++】在字符串中找出连续最长的数字串含-号 【2022 Q4 | 100分】 jidanza66689的博客 73
华为OD机试 C++ OD笔试题 在字符串中找出连续最长的数字串含-号:请在一个字符串中找出连续最长的数字串，并返回这个数字串。如果...

“相关推荐”对你有帮助么？

😞 非常没帮助

😐 没帮助

😬 一般

😊 有帮助

😄 非常有帮助