程博文 签名没有设置
关注 私信

242 原创 2309 周排名 5931 总排名 41万+ 访问 等级
3241 积分 2576 粉丝 145 点赞 968 评论 1342 收藏
勋章:
私信 已关注
搜博主文章
热门文章
华为OD机试真题2023（JAVA） 22905
华为OD机试（JAVA）真题列 16806
华为OD机试（JAVA）真题2022版 13367
【华为OD机试真题 JAVA】题格子游戏 6019
vdos命令报错报错无法挂载设置条（使用nginx解决） 5689
分类专栏
华为OD机试2023... 订阅 216篇
华为OD机试2022... 订阅 69篇
华为OD机试2022... 订阅 52篇
JPA 12篇
java 1篇
love 5篇
最新评论
华为OD机试真题2023（JAVA）
我.的bug: 收藏了，这题的文件夹吧怎麼
【亮分】【华为OD机试真题2023 JAVA...
一定，考试分数跟自己气色会很大比例
【亮分】【华为OD机试真题2023 JAVA...
m0_55384712: 这题不得了Map值转序后输出即
【亮分】【华为OD机试真题2023 JAVA...
哈哈哈，那的，那也觉得这个算法有问题，但是不得不佩人家牛了那么！
【亮分】【华为OD机试真题2023 JAVA...
哈哈哈，比喜欢不转换数字也是挺棒的！
您感觉自己最适合哪一博客详情页？
懂的不推荐 不推荐 一般般 推荐 强烈推荐
最新文章
【亮分】【华为OD机试真题2023 JAVA】最多几个直角三角形
Spring JPA @CreatedDate和@LastModifiedDate
JPA——Date和国之Calendar
2023
10月 2篇 11月 7篇 12月 3篇
2022年 192篇 2021年 14篇

目录

华为OD机试真题2023年度题目题库全
简单的解压缩算法
题目描述：
现需要实现一种算法，能将一组压缩字符串还原成原始字符串，还原规则如下：
输入描述：
输出描述：
补充说明：
示例1
输入：
输出：
说明：
示例2
输入：
输出：

【华为OD机试真题2023 JAVA】简单的解压缩算法

分享 程博文 发表于 2023-01-19 17:05:58 发布 2815 收藏 5 版权
分类专栏：华为OD机试2023（JAVA） 文章标签：java 算法 华为

广告 华为OD机试2023（...） 专题收录该内容 ¥59.90 ¥69.00 750 订阅 216 浏览量 已订阅 超级会员免费看
订阅专栏后可以享受以下权益：

华为OD机试真题，2023年度机试题库全覆盖，刷题指南点这里

简单的解压缩算法

知识点栈
时间限制：1s 空间限制：256MB 限定语言：不限

题目描述：

现需要实现一种算法，能将一组压缩字符串还原成原始字符串，还原规则如下：

1、字符后面加数字N，表示重复字符N次。例如：压缩内容为A3，表示原始字符串为AAA。

2、花括号中的字符串加数字N，表示花括号中的字符串重复N次。例如：压缩内容为{AB3}，表示原始字符串为ABABABB。

3、字符加N和花括号后面加N，支持任意的嵌套，但括号规格要求，例如：压缩内容可以{A3B1C}3}3。

输入描述：

输入一行压缩后的字符串

输出描述：

输出压缩前的字符串

补充说明：

输入保证，数字不会为0，花括号中的内容不会为空，保证输入的都是合法有效的压缩字符串
输入输出字符串仅为大小写
输入的字符串长度为范围1, 10000]
输出的字符串长度为范围1, 1000000]
数字N范围1, 10000]

示例1

输入：

```
A3
```

输出：

```
AAA
```

说明：

A3代表A字符重复3次

示例2

输入：

```
(A3B1C)3)3
```

输出：

```
AAABCCCAAABCCCAAABCCC
```

说明：

(A3B1C)3)代表A字符重复3次，B字符重复1次，花括号中的C字符重复3次，最外层花括号中的AAABCCC重复3次

解题思路：

例如：{A3B1C}3}2{Y}2L4}3

使用双端队列deque来放置字符串，使用temp字来放置临时字符串，isHasBrace代表是有大括号需要处理，初始化false

1、从头开始遍历，遇到“{”，isHasBraceefalse且temp字为空，则deque= {}；
最终遍历，直到第二个“}”，isHasBraceefalse但是temp非空值，则进行处理后放入队列中，deque= {{，AAAB，{}；

2、遇到字“}”，temp字为“C”，则deque= {{，AAAB，CCC，{}，isHasBrace是true，所以先处理完temp字放入队列中，deque= {{，AAAB，{，C}，isHasBrace是true；

3、遇到字“{”，因为isHasBrace=true，需要处理大括号：遇到队列，发现括号是最近有一个C，此时temp字=3，处理完放入队列中：deque= {{，AAAB，CCC，{}，isHasBrace是true；

4、遇到字“}”，temp字=“x2Y”，因为isHasBrace是true，所以先处理完temp字放入队列中，deque= {{，AAAB，CCC，{，xxY}，isHasBrace是true；

5、遇到字“L”，因为L是字符且isHasBrace=true，需要处理大括号：遇到队列，发现括号是最近是xxY，此时temp字=2，处理完放入队列中：deque= {{，AAAB，CCC，xxYxxY}，isHasBrace是false；

6、遇到字“}”，temp字=“L4”，因为isHasBrace=false，所以先处理完temp字放入队列中，deque= {{，AAAB，CCC，xxYxxY，LLLL}，isHasBrace是true；

7、遇到字符3，因为isHasBrace=true，需要处理大括号：遇到队列，发现括号是最近是AAABCCCxxYxxYLLLL，此时temp字=3，处理完放入队列中：deque= {AAABCCCxxYxxYLLLLAAABCCCxxYxxYLLLLAAABCCCxxYxxYLLLL}，
最终从头输出队列为：AAABCCCxxYxxYLLLLAAABCCCxxYxxYLLLLAAABCCCxxYxxYLLLL

代码：

```
public class Main{

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        String string = sc.nextLine();

        int len = string.length();
        Deque<String> strDeque = new ArrayDeque<>();
        String tempStr = "";// 字符序列
        boolean isHasBraces = false;// 是否有大括号需要处理
        for(int i=0; i<len; i++){
            char c = string.charAt(i);
            if(c == '{'){// 遇到左括号
                if(isHasBraces){// 有大括号需要处理
                    handleBraces(tempStr, strDeque);
                    isHasBraces = false;
                }else if(!tempStr.isEmpty()){
                    strDeque.addLast(handle(tempStr));
                }
                strDeque.addLast(String.valueOf(c));
                tempStr = "";
            }else if(c == '}'){
                strDeque.addLast(tempStr);
                isHasBraces = true;
                handleBraces() strDeque);
                strDeque.pollLast(), strDeque);
            }else {
                strDeque.addLast(handle(strDeque.pollLast()));
            }
            isHasBraces = true;
            tempStr = "";
        }else {
            if(isHasBraces && Character.isLetter(c)){
                handleBraces(tempStr, strDeque);
                isHasBraces = false;
                tempStr = "";
            }
            tempStr += c;
        }
        if(i == len - 1){ //最后一个字符
            if(isHasBraces){
                handleBraces(tempStr, strDeque);
            }else {
                strDeque.addLast(handle(tempStr));
            }
        }
    }

    String res = "";
    while (strDeque.size() != 0){
        res += strDeque.pollFirst();
    }

    System.out.println(res);
}

    /**
     * 处理大括号里面的字符串
     * @param strDeque
     * @return
     */
    public static String getBraces(Deque<String> strDeque){

        List<String> list = new ArrayList<>();
        while (!strDeque.peekLast().equals("{")){ // 直至碰到下一个左括号
            list.add(strDeque.pollLast());
        }
        strDeque.pollLast();// 注意弹出左括号
        Collections.reverse(list);// 因为是从尾部开始往队列放置数据

        String res = "";
        for(String s : list){
            res += s;
        }

        return res;
    }

    /**
     * 处理大括号里面的内容
     * @param tempStr        当前字符的数字
     * @param strDeque       字符队列
     */
    public static void handleBraces(String tempStr, Deque<String> strDeque){

        int num = Integer.valueOf(tempStr);
        String temp = "";
        String strInBraces = getBraces(strDeque);
        for(int j=0; j<num; j++){
            temp += strInBraces;
        }
        strDeque.addLast(temp);
    }

    /**
     * 处理没有括号的字符串
     * @param str
     * @return
     */
    public static String handle(String str){

        int len = str.length();

        String res = "";
        String tempStr = "";// 字符序列
        String tempNum = "";// 数字序列字符串（处理多位数）
        for(int i=0; i<len; i++){
            char c = str.charAt(i);
            if(Character.isDigit(c)){ // 此时是数字
                tempNum += c;// 数字字符串（处理多位数）
            }else {// 此时是字母
                if(!tempNum.isEmpty()){ // 有字符串不为空说明需要处理
                    int num = Integer.valueOf(tempNum);// 字符重复的次数
                    for(int j=0; j<num; j++){
                        res += tempStr;// 重复多少次就拼接多少次
                    }
                    tempStr = "";// 处理完需要重复以后数据下一个字符的统计
                    tempNum = "";// 数字一样重置
                }
                tempStr += c;// 字符拼接
            }
            if(i == len - 1){ // 不是以字母结尾的一位
                if(!tempNum.isEmpty()){ // 有数字字符串不为空的处理
                    int num = Integer.valueOf(tempNum);
                    for(int j=0; j<num; j++){
                        res += tempStr;
                    }
                }else {
                    res += tempStr;        // 为空时直接拼接字符串（处理单字符串的情况）
                }
            }
        }

        return res;
    }
}
```


题 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 4097人正在系统学习中

【华为OD机试真题 python】简单的解压缩算法 【2022 Q4 | 200分】 无夜叶 2021年博客 475
华为OD机试 Python OD机试 简单的解压缩算法 题目描述 现需要实现一种算法，能将一组压缩字符串还原成原始字符串，还原规则如下：

Printer Queue算法（华为）打印任务排序 PQ3125|Golang|正程 01-20
问题描述 Printer Queue（打印队列） PQ3125 打印机调度打印问题 这是一道ACM算法题，上面的两个是求打印时间，还有一种是求打印...

评论 3 条 写评论

核心代码

```
核心代码
int begin = a.lastIndexOf('}')+1;
int begin = a.lastIndexOf('{');
int last = a.indexOf('}');
String temp = "";
for (int i = last+1; i < a.length(); i++) {
    char b = a.charAt(i);
    if(b>='0'&&b<='9'){
        temp += b;
    }else{
        break;
    }
}
s = a.substring(0,begin)+cheng(parseString(a.substring(begin+1,last)),tem
}
```

nowl 测评

抽题优化算法（matlab实现） 05-30
抽题，世界上已建立了数目繁杂的世界，物理、建筑、商林等相关的数学模型 其中作物模型一直是植物科学模型研究的重点之一，国外...

【华为OD机试真题 Python】简易压缩算法 A_D_LZ_A 3小时前 666
题目描述 现需要实现一种算法，能将一组压缩字符串还原成原始字符串，规则如下：1、字符后面加数字N，表示重复字符N次。例如：压缩内容为A3，表示原始字符串为AAA。2、花括号中的字符串加数字N，表示花括号中的字符串重复N次。例如：压缩内容为{AB3}，表示原始字符串为ABABABB。3、字符加N和花括号后面加N，支持任意的嵌套，但括号规格要求，例如：压缩内容可以{A3B1C}3}3。

【华为机试真题 Python实现】 字符串压缩解压缩 不太系统的菜鸡 1484
文章目录题目描述示例 1参考代码 题目描述 将一段压缩后的字符串解压缩，并且按字母出现频次倒序输出一个数字...

华为OD机试 - 改结随机 (Java & JS & Python) qq_1262269博客 2011
2022 Q4 三页真题Java、JavaScript、Python，三个语言

华为OD机试真题2023 JAVA 万家致富的小布 qq_34455338的博客 1669
不要脸的小布 刚还完二空房的小布，刚还完二空房的小布，h空间限制：256MB限定语言：不限 题目描述：某古镇主要建了一大片果园，field[i] 表示...

华为od机考真题-水果搬运问题 VAN 292
华为OD机考真题

2021校招算法题每日学习（4） weixin_44979165的博客 137
海伯尔Dijkstra，不要用图论图 也无向 有向图 bellman-ford

P1879 玉米田Corn Fields题解 Musk melon 285
洛谷 P1879 题解 - P1879 玉米田Corn Fields 状态压缩dp实现 一维状态转移方程 - 状态压缩模版分值MT(N为1 ≤ M ≤ 12, 1 ≤ N ≤ 12...

华为OD机试真题JAVA实现【改编版交流群】 qfq_12262的博客 206
如果你的题目跟华为有关，你的题目可能会有帮助，欢迎加入华为OD交流群 - 作者公众号Bugfree 有问题一个群的打字速度，请字符串的长度为数字最大的字符...

华为OD机试 - 简单的解压缩算法 (Java & JS & Python) qfq_12262的博客 1617
2022 Q4 三页真题Java、Python、JavaScript，考察：模拟题

华为OD机试（2022&2023）专栏介绍 + 真题目录 热门推荐 qfq_12262的博客 275+
华为OD机试真题 (Java & JavaScript & Python)

农场施肥 (华为OD机试 2023 Q1考试题 A卷) weixin_54797169的博客 15
题目描述 (农场施肥) 某古镇主要建了一大片果园，field[i] 表示不同果林的面积，单位 (m^2)。现在要为所有的果林施肥总必须是 x 天...

Leetcode 面试题 01.06.字符串压缩（Compressing String LCCI） 3a7f7b22的博客 98
Leetcode 面试题 01.06.字符串压缩（Compressing String LCCI）字符串压缩，利用字符重复出现的次数，编写一种方法，实...

【100%通过率】华为OD机试真题 C++ 实现 【最长回文字符串】 【2023 Q1 | 100...... 最新发布 MISAYAONEFOREVER 2691
给一个包括大小写字母的字符串，请用这些字母组成一个最长的回文串，若有多个最长的，返回其中字典序最小的回文串，字母...

[C/C++]OJ练习题：压缩字符串&压缩字符串 (找词频) 春眠白日枕花眠 1206
小一想出一个新的压缩字符串的方法：将连续出现的字母缩写成一个数字，若该字母出现的频率大于...

字符压缩算法15.11 weixin_43952190的博客 195
字符串压缩算法 有一种简单的字符串压缩算法，对于字符串中连续出现的同一字符，用该字符加上连续出现的次数来表示（连续出现次...

【Leetcode刷题篇】leetcode443.压缩字符串 fdnd_programmerzxy的博客 202
题目：给定一组字符，使用原地算法将其压缩。压缩后的长度必须始终小于或等于原数组长度。 数组的每个元素应该是长度为1 的字符...

"相关推荐"对你有帮助么？
非常实用 有帮助 一般 无帮助 非常无帮助

关于我们 招聘合作 广告服务 开发平台 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00