华为OD机试 - 报文重排序 (Java & JS & Python)

原创 伏城之外 已于 2023-06-04 14:15:22 修改 1745 收藏 3 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

对报文进行重传和重排序是常用的可靠性机制，重传缓中区内有一定数量的子报文，每个子报文在原始报文中的顺序已知，现在需要恢复出原始报文。

输入描述

输入第一行为N，表示子报文的个数，0 < N ≤ 1000。
输入第二行为N个子报文，以空格分开，子报文格式为：

> 字符串报文内容+后缀顺序索引

字符串报文内容由[a-z,A-Z]组成，后缀为整型值，表示顺序。

顺序值唯一，不重复。

输出描述

输出恢复出的原始报文，按照每个子报文的顺序的升序排序恢复出原始报文，顺序后缀需要从恢复出的报文中删除掉

用例

输入	4 rolling3 stone4 like1 a2
输出	like a rolling stone
说明	4个子报文的内容分别为 "rolling"，"stone"，"like"，"a"，顺序值分别为3，4，1，2，按照顺序值升序并删除顺序后缀，得到恢复的原始报文："like a rolling stone“

输入	8 gifts6 and7 Exchanging1 all2 precious5 things8 kinds3 of4
输出	Exchanging all kinds of precious gifts and things
说明	无

题目解析

考察字符串操作。

这题解析“字符串报文内容”、"后缀顺序索引"时，我使用了正则匹配的捕获组。

2023.06.04

本题需要考虑下后缀顺序索引不连续的情况，即不是严格的0~N-1序号

```
1 import java.util.ArrayList;
```

```java
import java.util.Scanner;
import java.util.StringJoiner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Main {
  static class Word {
    int id;
    String content;

    public Word(int id, String content) {
      this.id = id;
      this.content = content;
    }
  }

  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    int n = Integer.parseInt(sc.nextLine());
    String[] arr = sc.nextLine().split(" ");

    System.out.println(getResult(n, arr));
  }

  public static String getResult(int n, String[] arr) {
    ArrayList<Word> ans = new ArrayList<>();

    Pattern pattern = Pattern.compile("([a-zA-Z]+)(\\d+)");

    for (String s : arr) {
      Matcher matcher = pattern.matcher(s);
      if (matcher.find()) {
        String content = matcher.group(1);
        int i = Integer.parseInt(matcher.group(2)) - 1;
        ans.add(new Word(i, content));
      }
    }

    ans.sort((a, b) -> a.id - b.id);
```

```
42
43      StringJoiner sj = new StringJoiner(" ");
44      for (Word an : ans) {
45        sj.add(an.content);
46      }
47      return sj.toString();
48    }
49  }
```