

Java方向编程题答案

第八周

day45

810 乒乓球筐

链接: <https://www.nowcoder.com/questionTerminal/bb4f1a23dbb84fd7b77be1fbe9eaaf82>

【题目解析】:

题目比较明确, 注意审题即可.

【解题思路】:

借助 map 统计出每个盒子中的每种球的类别和数目. 然后遍历其中的一个 map 和另外一个 map 进行对比即可. 具体细节参考代码注释.

```
// 基本思路
// 借助 哈希表 保存每个盒子中的每种球的类别和数目.
// 然后遍历其中的一个 map 和另外一个 map 进行对比即可.

import java.util.*;

public class Main{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        while(sc.hasNext()){
            String boxA=sc.next();
            String boxB=sc.next();
            // 创建两个 hash 表. key 为球的种类, value 为球的数量
            HashMap<String,Integer> hashMapBoxA=new HashMap<String,Integer>();
            HashMap<String,Integer> hashMapBoxB=new HashMap<String,Integer>();
            // 初始化 BoxA
            for(int i=0;i<boxA.length();i++){
                if(hashMapBoxA.containsKey(boxA.charAt(i)+"")){
                    hashMapBoxA.put(boxA.charAt(i)+"",hashMapBoxA.get(boxA.charAt(i)+"")+1);
                }else{
                    hashMapBoxA.put(boxA.charAt(i)+"",1);
                }
            }
            // 初始化 BoxB
            for(int i=0;i<boxB.length();i++){
                if(hashMapBoxB.containsKey(boxB.charAt(i)+"")){
                    hashMapBoxB.put(boxB.charAt(i)+"",hashMapBoxB.get(boxB.charAt(i)+"")+1);
                }else{
                    hashMapBoxB.put(boxB.charAt(i)+"",1);
                }
            }
            // 遍历 BoxB, 和 BoxA 中的元素进行对比. flag 表示是否是满足包含关系
            int flag=0;
```

```

        for(String key:hashMapBoxB.keySet()){
            if(!hashMapBoxA.containsKey(key)){
                flag=1;
                break;
            }else{
                if(hashMapBoxA.get(key)<hashMapBoxB.get(key)){
                    flag=1;
                    break;
                }
            }
        }
        if(flag==1){
            System.out.println("No");
        }else{
            System.out.println("Yes");
        }
    }
}
}

```

36851 查找兄弟单词

链接: <https://www.nowcoder.com/questionTerminal/03ba8aeeef73400ca7a37a5f3370fe68>

【题目解析】：

兄弟单词的含义: 两个单词不同, 长度相同, 但是构成的字母顺序不相同

输入描述: 先输入字典中单词的个数n, 再输入n个单词作为字典单词。再输入一个单词, 查找其在字典中兄弟单词的个数m 再输入数字k

输出描述: 根据输入, 输出查找到的兄弟单词的个数m 然后输出查找到的兄弟单词的第k个单词。注意!! 一定要关注最终返回的第 k 个单词这样的序号, 不能搞错。

题意说的是 "字典", 因此要将里面的单词按照字典序排序~否则得到的 k 会存在问题

【解题思路】：

1. 将字典中的单词先放到 vector 中
2. 将 vector 进行排序
3. isBrother 函数依次判定每个输入的单词是否是兄弟单词
4. 判定兄弟单词的规则是
 1. 先判定长度
 2. 如果长度相同, 再看是否是完全相同(完全相同不算兄弟)
 3. 然后将两个单词排序, 排序相同才是真兄弟单词。

```

import java.util.Arrays;
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;

```

```

public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        while(in.hasNext()){
            int num = in.nextInt();
            String[] s = new String[num];
            int count = 0;
            for(int i = 0; i < num; i++){
                s[i] = in.next();
            }
            String key = in.next();
            char[] keyChar = key.toCharArray();
            Arrays.sort(keyChar);
            int no = in.nextInt();//第几个
            ArrayList<String> list = new ArrayList<String>();
            for(int i = 0; i < num; i++){
                int c = check(key, s[i], keyChar);
                count += c;
                if(c == 1)
                    list.add(s[i]);
            }
            System.out.println(count);
            Collections.sort(list);
            if(count >= no)
                System.out.println(list.get(no-1));
        }
    }

    // 核心的比较函数。keyChar 在调用的时候就已经排好序了。
    private static int check(String key, String word, char[] keyChar){
        // 就是将两个单词按字符排序后看相不相同就行了。
        if(key.equals(word) || key.length() != word.length())
            return 0;
        char[] wordChar = word.toCharArray();
        Arrays.sort(wordChar);
        return Arrays.equals(keyChar, wordChar)?1:0;
    }
}

```