

题目描述

电视剧《分界线》里面有一个片段，男主为了向警察透露案件细节，且不暴露自己，于是将报刊上的字减下来，剪拼成匿名信。
现在又一名举报人，希望借鉴这种手段，使用英文报刊完成举报操作。
但为了增加文章的混淆度，只需满足每个单词中字母数量一致即可，不关注每个字母的顺序。
解释：单词on允许通过单词no进行替代。
报纸代表newspaper，匿名信代表anonymousLetter，求报纸内容是否可以拼成匿名信。

输入描述

第一行输入newspaper内容，包括1-N个字符串，用空格分开
第二行输入anonymousLetter内容，包括1-N个字符串，用空格分开。

newspaper和anonymousLetter的字符串由小写英文字母组成，且每个字母只能使用一次；
newspaper内容中的每个字符串字母顺序可以任意调整，但必须保证字符串的完整性（每个字符串不能有多余字母）
 $1 < N < 100$,
 $1 \leq \text{newspaper.length}, \text{anonymousLetter.length} \leq 10^4$

输出描述

如果报纸可以拼成匿名信返回true，否则返回false

用例

输入	ab cd ab
输出	true
说明	无

输入	ab ef aef
输出	false
说明	无

输入	ab bcd ef cbd fe
输出	true
说明	无

输入	ab bcd ef cd ef
输出	false
说明	无

题目解析

用例1的意思是：
报纸上有两个单词：ab、cd，而要写的匿名信需要一个单词ab，因此可以直接使用报纸上的单词ab，所以可以写出匿名信。

用例2的意思是：
报纸上有两个单词：ab、ef，而要写的匿名信需要一个aef，根据题目意思

只需满足每个单词中字母数量一致即可

如果想用报纸上的单词，代替匿名信上的单词，则这两个单词的字母数量必须一致。
因此，对于匿名信单词aef，有三个字母，而报纸上没有有三个字母的单词，因此输出false。

题目解析

用例1的意思是：

报纸上有两个单词：ab、cd，而要写的匿名信需要一个单词ab，因此可以直接使用报纸上的单词ab，所以可以写出匿名信。

用例2的意思是：

报纸上有两个单词：ab、ef，而要写的匿名信需要一个aef，根据题目意思

只需满足每个单词中字母数量一致即可

如果想用报纸上的单词，代替匿名信上的单词，则这两个单词的字母数量必须一致。

因此，对于匿名信单词aef，有三个字母，而报纸上没有有三个字母的单词，因此输出false。

我增加一个自测用例，说明下面这个特点

不关注每个字母的顺序。单词on允许通过单词no进行替代。

比如，报纸单词ba、cd，匿名信需要单词ab，而题目说

不关注每个字母的顺序

因此匿名信的单词ab可以用报纸的单词ba代替，因此输出true。

本题需要关注下数量级：1 <= newspaper.length, anonymousLetter.length <= 10⁴

因此双重for的O(n²)时间复杂度会高达一亿次循环，会超时。

我的策略如下，统计出newspaper中每个单词出现的次数到count对象中，统计前，对每个单词进行字典序排序^Q，这样就可以忽略字母顺序了。

比如，用例1的newspaper会统计出count: { "ab": 1, "cd": 1 }，这个时间复杂度是O(n)

然后，再遍历anonymousLetter每个单词，并对单词进行字典序排序，忽略字母顺序，然后用排序后单词去count中找，如果count[letter] > 0，则可以找到，然后count[letter]--

遵循题目意思：每个字母只能使用一次

如果count[letter]不存在，或者count[letter]===0，则说明匿名信中的某个单词在报纸单词中找不到替代，因此可以直接返回false。

如果全部都可以找到，则返回true。

这个过程的时间复杂度是O(n)。

因此总的时间复杂度是O(n)。

JavaScript算法源码

```
1  /* JavaScript Node ACM模式, 控制台输入获取 */
2  const readline = require("readline");
3
4  const rl = readline.createInterface({
5    input: process.stdin,
6    output: process.stdout,
7  });
8
9  const lines = [];
10 rl.on("line", (line) => {
11   lines.push(line);
12 });
13 if (lines.length === 2) {
14   const newspaper = lines[0].split(" ");
15   const anonymousLetter = lines[1].split(" ");
16   console.log(getResult(newspaper, anonymousLetter));
17 }
18 lines.length = 0;
19 }
20 });
21
22 function getResult(newspaper, anonymousLetter) {
23   const count = {};
24   for (let str of newspaper) {
25     str = [...str].sort().join("");
26     count[str] ? count[str]++ : (count[str] = 1);
27   }
28
29   let flag = true;
30   for (let str of anonymousLetter) {
31     str = [...str].sort().join("");
32     if (count[str] > 0) {
33       count[str]--;
34     } else {
35       flag = false;
36       break;
37     }
38   }
```

JavaScript算法源码

```
1  /* JavaScript Node ACM模式 控制台输入获取 */
2  const readline = require("readline");
3
4  const rl = readline.createInterface({
5    input: process.stdin,
6    output: process.stdout,
7  });
8
9  const lines = [];
10 rl.on("line", (line) => {
11   lines.push(line);
12
13   if (lines.length === 2) {
14     const newspaper = lines[0].split(" ");
15     const anonymousLetter = lines[1].split(" ");
16     console.log(getResult(newspaper, anonymousLetter));
17
18     lines.length = 0;
19   }
20 });
21
22 function getResult(newspaper, anonymousLetter) {
23   const count = {};
24   for (let str of newspaper) {
25     str = [...str].sort().join("");
26     count[str] ? count[str]++ : (count[str] = 1);
27   }
28
29   let flag = true;
30   for (let str of anonymousLetter) {
31     str = [...str].sort().join("");
32     if (count[str] > 0) {
33       count[str]--;
34     } else {
35       flag = false;
36       break;
37     }
38   }
39
40   return flag;
41 }
```

Java算法源码

```
1  import java.util.Arrays;
2  import java.util.HashMap;
3  import java.util.Scanner;
4
5  public class Main {
6    public static void main(String[] args) {
7      Scanner sc = new Scanner(System.in);
8
9      String[] newspaper = sc.nextLine().split(" ");
10     String[] anonymousLetter = sc.nextLine().split(" ");
11
12     System.out.println(getResult(newspaper, anonymousLetter));
13   }
14
15   public static boolean getResult(String[] newspaper, String[] anonymousLetter) {
16     HashMap<String, Integer> count = new HashMap<>();
17     for (String str : newspaper) {
18       String newStr = strSort(str);
19       count.put(newStr, count.getOrDefault(newStr, 0) + 1);
20     }
21
22     boolean flag = true;
23     for (String str : anonymousLetter) {
24       String newStr = strSort(str);
25
26       if (count.containsKey(newStr) && count.get(newStr) > 0) {
27         count.put(newStr, count.get(newStr) - 1);
28       } else {
29         flag = false;
30         break;
31       }
32     }
33
34     return flag;
35   }
36
37   public static String strSort(String str) {
38     char[] cArr = str.toCharArray();
39     Arrays.sort(cArr);
40     return new String(cArr);
41   }
42 }
```

Python算法源码

Java算法源码

```
1 import java.util.Arrays;
2 import java.util.HashMap;
3 import java.util.Scanner;
4
5 public class Main {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8
9         String[] newspaper = sc.nextLine().split(" ");
10        String[] anonymousLetter = sc.nextLine().split(" ");
11
12        System.out.println(getResult(newspaper, anonymousLetter));
13    }
14
15    public static boolean getResult(String[] newspaper, String[] anonymousLetter) {
16        HashMap<String, Integer> count = new HashMap<>();
17        for (String str : newspaper) {
18            String newStr = strSort(str);
19            count.put(newStr, count.getOrDefault(newStr, 0) + 1);
20        }
21
22        boolean flag = true;
23        for (String str : anonymousLetter) {
24            String newStr = strSort(str);
25
26            if (count.containsKey(newStr) && count.get(newStr) > 0) {
27                count.put(newStr, count.get(newStr) - 1);
28            } else {
29                flag = false;
30                break;
31            }
32        }
33
34        return flag;
35    }
36
37    public static String strSort(String str) {
38        char[] cArr = str.toCharArray();
39        Arrays.sort(cArr);
40        return new String(cArr);
41    }
42 }
```

Python算法源码

```
1 # 输入获取
2 newspaper = input().split()
3 anonymousLetter = input().split()
4
5
6 # 算法源码
7 def getResult(newspaper, anonymousLetter):
8     count = {}
9     for s in newspaper:
10         s = "".join(sorted(s))
11         if count.get(s) is None:
12             count[s] = 1
13         else:
14             count[s] += 1
15
16     flag = True
17     for s in anonymousLetter:
18         s = "".join(sorted(s))
19         if count.get(s) is not None and count[s] > 0:
20             count[s] -= 1
21         else:
22             flag = False
23             break
24
25     return str(flag).lower()
26
27
28 # 调用算法
29 print(getResult(newspaper, anonymousLetter))
```