

题目描述

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母的元素长度乘积的最大值。
如果有符合条件的两个元素，返回0。

输入描述

输入为一个半角逗号分隔的小写字母串的数字，2 <= 数组长度 <= 100，0 < 字符串长度 <= 50。

输出描述

两个没有相同字母的元素 长度乘积的最大值。

用例

输入	wcdybenjh,kap,ist,kadggf
输出	14
说明	数组中有5个元素。 wcdyben与jh无相同的字母，满足条件，wcdyben的长度为7，jh的长度为2，乘积为14（7*2）。 wcdyben与kap、ist、kadggf含有相同的字母，不满足条件。 kap与istd、kadggf含有相同的字母，不满足条件。 ist与kadggf含有相同的字母，不满足条件。 因此，输出为14。

题目解析

交题问题。

JavaScript算法源码

```
1 // 题目描述
2 const readline = require('readline');
3
4 const rl = readline.createInterface({
5   input: process.stdin,
6   output: process.stdout,
7 });
8
9 rl.on('line', (line) => {
10   const arr = line.split('').filter(x => x !== ' ');
11   console.log(getResult(arr));
12 });
13
14 function getResult(arr) {
15   let ans = 0;
16
17   const sets = arr.map(s => new Set([...s]));
18
19   for (let i = 0; i < sets.length; i++) {
20     for (let j = i + 1; j < sets.length; j++) {
21       if (disjoint(sets[i], sets[j])) {
22         ans = Math.max(ans, arr[i].length * arr[j].length);
23       }
24     }
25   }
26
27   return ans;
28 }
29
30 function disjoint(set1, set2) {
31   for (let c of set1) {
32     if (set2.has(c)) return false;
33   }
34   return true;
35 }
```

Java算法源码

```
1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashMap;
4 import java.util.Scanner;
5
6 public class Main {
7
8   public static void main(String[] args) {
9     Scanner sc = new Scanner(System.in);
10     String[] strings = sc.nextLine().split(",");
11     System.out.println(getResult(strings));
12   }
13
14   public static int getResult(String[] arr) {
15     ArrayList<HashSet<Character>> list = new ArrayList<>();
16
17     for (String s : arr) {
18       HashSet<Character> set = new HashSet<>();
19       for (char c : s.toCharArray()) set.add(c);
20       list.add(set);
21     }
22
23     let ans = 0;
24
25     for (int i = 0; i < list.size(); i++) {
26       HashSet<Character> s = list.get(i);
27       for (int j = i + 1; j < list.size(); j++) {
28         HashSet<Character> t = list.get(j);
29         if (Collections.disjoint(s, t)) {
30           ans = Math.max(ans, s.size() * t.size());
31         }
32       }
33     }
34
35     return ans;
36   }
37 }
```

Python算法源码

```
1 # 题目描述
2 arr = input().split(',')
3
4 # 题目解析
5 def getResult():
6     sets = list(map(lambda x: set(x), arr))
7
8     ans = 0
9
10    for i in range(len(sets)):
11        for j in range(i+1, len(sets)):
12            if sets[i].isdisjoint(sets[j]):
13                ans = max(ans, len(sets[i]) * len(sets[j]))
14
15    return ans
16
17 # 测试用例
18 print(getResult())
```