

Java-字符串-一群大雁往南飞

题目描述：

一群大雁往南飞，给定一个字符串记录地面上的游客听到的大雁叫声，请给出叫声最少由几只大雁发出。具体的：

1. 大雁发出的完整叫声为"quack"，因为有多只大雁同一时间嘎嘎作响，所以字符串中可能会混合多个 "quack"。
2. 大雁会依次完整发出 "quack"，即字符串中'q','u','a','c','k' 这 5 个字母按顺序完整存在才能计数为一只大雁。如果不完整或者没有按顺序则不予计数。
3. 如果字符串不是由'q','u','a','c','k'字符组合而成,或者没有找到一只大雁,请返回 -1。

输入描述：

一个字符串，包含大雁 quack 的叫声。1 <= 字符串长度 <= 10000

字符串中的字符只有'q','u','a','c','k'

输出描述：

大雁的数量

补充说明：

示例 1

输入：

quackquack

输出：

1

说明：

一只大雁叫了两次

示例 2

输入：

quqackuack

输出：

2

说明：

最少需要两只大雁分别叫一次，第一只雁子 quqackuack ,第二只雁子 quqackuack

示例 3

输入:

quackquook

输出:

-1

说明:

给出的字符串不是 "quack"的有效组合。

```
import java.util.Scanner;
```

```
import java.util.LinkedList;
```

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

```
// 注意类名必须为 Main, 不要有任何 package xxx 信息
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Scanner in = new Scanner(System.in);
```

```
        String s = in.next();
```

```
        System.out.println(function(s));
```

```
    }
```

```
    public static int function(String s){
```

```
        LinkedList<Integer> linkedList = new LinkedList<>();
```

```
        int num1 = 0, num2 = 0, num3 = 0;
```

```
        ArrayList<Integer[]> arrayList = new ArrayList<>();
```

```
// 遍历
```

```
for (int i = 0; i<s.length();i++) {
```

```
    // switch 判断
```

```
    switch(s.charAt(i)) {
```

```
        case 'q':
```

```
            linkedList.add(i);
```

```
            break;
```

```
        case 'u':
```

```
            if(num1+1<=linkedList.size()) num1++;
```

```
            break;
```

```
        case 'a':
```

```
            if(num2+1<=num1) num2++;
```

```
            break;
```

```
        case 'c':
```

```
            if(num3+1<=num2) num3++;
```

```
            break;
```

```
        case 'k':
```

```
            if(num3>=1) {
```

```
                arrayList.add(new Integer[]
```

```
{linkedList.removeFirst(),i});
```

```
                num1--;
```

```
num2--;
```

```
num3--;
```

```
}
```

```
break;
```

```
default:
```

```
return -1;
```

```
}
```

```
}
```

```
if (0 == arrayList.size()) return -1;
```

```
int moo = 1;
```

```
for (int i = 0; i<arrayList.size();i++) {
```

```
    int mooCount = 1;
```

```
    for (int j = i+1;j<arrayList.size();j++) {
```

```
        if(arrayList.get(i)[1] >= arrayList.get(j)[0]) {
```

```
            mooCount++;
```

```
        }
```

```
    }
```

```
    moo = Math.max(moo,mooCount);
```

```
}
```

```
return moo;
```

}

}