

垃圾信息拦截

知识点图

时间限制：1s 空间限制：256MB 限定语言：不限

题目描述：

大众对垃圾短信深恶痛绝，希望能对垃圾短信发送者进行识别，为此，很多软件增加了垃圾短信识别机制。经分析，发现正常用户的短信通常具备交互性，而垃圾短信往往都是大量单向的短信，按照如下规则进行垃圾短信识别：

本题中，发送者A符合以下条件之一的，则认为A是垃圾短信发送者：

- * A发送短信的接收者中，没有发过短信给A的人数 $L > 5$;
- * A发送的短信数 - A接收的短信数 $M > 10$;
- * 如果存在X，A发送给X的短信数-A接收到X的短信数 $N > 5$;

输入描述：

第一行为条目数目，接下来几行是具体的条目，每个条目，是一对ID，第一个数字是发送者ID，后面的数字是接收者ID，中间空格隔开，所有的ID都为无符号整型，ID最大值为100；

同一个条目中，两个ID不会相同（即不会自己给自己发消息）

最后一行为指定的ID

输出描述：

输出该ID是否为垃圾短信发送者，并且按序列输出 L M的值（由于N值不唯一，不需要输出）；
输出均为字符串。

示例1

输入：

```
15
12
13
14
15
16
17
18
19
1 10
1 11
1 12
1 13
1 14
14 1
1 15
1
```

输出：

```
true 13 13
```

说明：

true表示1是垃圾短信发送者，两个数字，代表发送者1对应的L和M值。true 13 13中间以一个空格分割。注意true是 **字符串输出**。

示例2

输入：

```
15
1 2
1 3
1 4
1 5
1 6
1 7
1 8
1 9
1 10
1 11
1 12
1 13
1 14
14 1
1 15
2
```

输出：

```
false 0 -1
```

解题思路：

本题主要是逻辑题，不是很难。需要注意的就是使用map对象来记录A发送给X的短信数和A接收到X的短信数N。

```
public class Main{

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        List<int[]> list = new ArrayList<>();
        for(int i=0; i<n; i++){
            int idFrom = sc.nextInt();
            int idTo = sc.nextInt();
            list.add(new int[]{ idFrom, idTo});
        }
    }
}
```

```

int id = sc.nextInt();          //A 的 ID

Set<Integer> sendMailPersons = new HashSet<>();    //接收 A 短信的人集合
Set<Integer> getMailPersons = new HashSet<>();     //发送短信给 A 的人集合
Map<Integer, Integer> sendMail = new HashMap<>();  //A 发送短信给 X 的数量
Map<Integer, Integer> getMail = new HashMap<>();   //A 接收 X 短信的数量
int sendMailsNum = 0;    //A 发送的短信数量
int getMailsNum = 0;     //A 接收的短信数量
for(int[] ints : list){
    if(ints[0] == id){    //发送者是 A
        sendMailPersons.add(ints[1]);
        sendMailsNum ++;
        sendMail.put( ints[1], sendMail.getDefault( ints[1], 0) + 1);
    }else if(ints[1] == id){    //接受者是 A
        getMailsNum ++;
        getMailPersons.add(ints[0]);
        getMail.put( ints[0], getMail.getDefault( ints[0], 0) + 1);
    }
}

boolean isLaji = false;
sendMailPersons.removeAll(getMailPersons);    //接收 A 短信的人的集合剔除发
送短信给 A 的人集合
int L = sendMailPersons.size();
int M = sendMailsNum - getMailsNum;
if(L > 5){
    isLaji = true;
}else if(M > 10){
    isLaji = true;
}else {
    for(Map.Entry<Integer, Integer> map : sendMail.entrySet()){
        int getId = map.getKey();
        int send = map.getValue();
        if(getMail.containsKey(getId)){
            if(send - getMail.get(getId) > 5){
                isLaji = true;
                break;
            }
        }
    }
}

System.out.println(isLaji + " " + L + " " + M);
}

```

