

报文回路

题目描述：

IGMP 协议中响应报文和查询报文，是维系组播通路的两个重要报文，在一条已经建立的组播通路中两个相邻的 HOST 和 ROUTER，ROUTER 会给 HOST 发送查询报文，HOST 收到查询报文后给 ROUTER 回复一个响应报文，以维持相互之间互通的关系，一旦这个关系断裂，那么这条组播通路就“异常”了。现通过某种手段，抓取到了 HOST 和 ROUTER 两者通讯的所有响应报文和查询报文，请分析该组播通路是否“正常”

输入描述：

第一行抓到的报文数量 C（C<=100），后续 C 行依次输入设备节点 D1 和 D2，表示从 D1 到 D2 发送了单向的报文，D1 和 D2 用空格隔开。

输出描述：

组播通路是否“正常”，正常输出 True，异常输出 False。

补充说明：

示例 1

输入：

```
5
1 2
2 3
3 2
1 2
2 1
```

输出：

```
True
```

说明：

示例 2

输入：

```
3
1 3
```

3 2

2 3

输出:

False

说明:

```
#include <iostream>
#include <vector>
#include <set>
#include <algorithm>
using namespace std;
bool getres(vector<vector<int>> &V)
{
    bool res=true;
    set<pair<int,int>> tool;
    vector<int> to_tool;
    vector<int> from_tool;
    int size=V.size();
    for(int i=0;i<size;++i)
    {
        pair<int,int> tmp;
        tmp.first=V[i][0];
        tmp.second=V[i][1];
        if(tool.count(tmp)==0)
        {
            tool.insert(tmp);
        }
    }
    set<pair<int, int>>::iterator ite=tool.begin();
    while(ite!=tool.end())
    {
        from_tool.push_back((*ite).first);
        to_tool.push_back((*ite).second);
        ite++;
    }
    sort(from_tool.begin(),from_tool.end());
    sort(to_tool.begin(),to_tool.end());
    int tool_size=tool.size();
    if(tool_size%2)return false;
    for(int i=0;i<tool_size;++i)
    {
        if(from_tool[i]!=to_tool[i])
        {
            res=false;
        }
    }
}
```

```

        break;
    }
}
return res;
}
int main()
{
    int len;
    cin >> len;
    vector<vector<int>> V(len, vector<int>(2));
    for (int i = 0; i < len; ++i)
    {
        int num;
        cin >> num;
        V[i][0] = num;
        cin >> num;
        V[i][1] = num;
    }
    if(getres(V))
    {
        cout<<"True"<<endl;
    }else{
        cout<<"False"<<endl;
    }
    return 0;
}

```