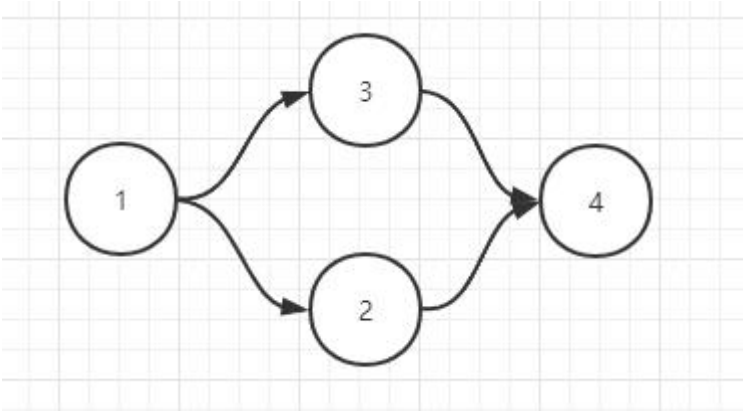
题目描述：
给定一个有向图，图中可能包含有环，图使用二维矩阵表示，每一行的第一列表示起始节点，第二列表示终止节点，如[0, 1]表示从 0 到 1 的路径。每个节点用正整数表示。求这个数据的首节点与尾节点，题目给的用例会是一个首节点，但可能存在多个尾节点。同时，图中可能含有环。如果图中含有环，返回[-1]。
说明：入度为 0 是首节点，出度为 0 是尾节点。



输入描述：
第一行为后续输入的键值对数量 N>=0，第二行为 2N 个数字。每两个为一个起点，一个终点。如：

输出描述：
输出一行头节点和尾节点。如果有多个尾节点，按从大到小的顺序输出。

补充说明：
如果图有环，输出为-1.
所有输入均合法，不会出现不配对的数据
示例
示例 1
输入：
4
0 1 0 2 1 2 2 3
输出：
0 3
说明：
示例 2
输入：
2
0 1 0 2
输出：
0 1 2
说明：
#include <iostream>

```
#include<unordered_map>
#include<vector>
#include<algorithm>
using namespace std;

int main() {
    int n;
    cin >> n;
    int a,b;
    unordered_map<int,int>in, out;
    for(int i=0;i<n;i++){
        cin >> a >> b;//a->b
        out[a]++;//a 出度+1
        in[b]++;//b 入度+1
    }
    //如果有节点入度为 0，则为首节点
    //如果有节点出度为 0，则为尾节点
    //如果都没有，则说明有环
    bool flag = false;
    for(auto &[d,cnt]:out){
        if(!in.count(d)){
            flag = true;
            cout << d << ' ';
            break;
        }
    }
    if(!flag)cout << -1 << endl;
    else{
        vector<int>res;
        for(auto &[d,cnt]:in){
            if(!out.count(d)){
                res.push_back(d);
            }
        }
        if(res.size()){
            sort(res.begin(),res.end());
            for(int&r:res)cout << r << ' ';
        }
        //else cout << -1;
    }
}
// 64 位输出请用 printf("%lld")
```