

Graph Traversal Application

In Degree : the number of edges enter the node .

Out Degree : the number of edges go out of the node .

Q

هيديك عدد من ال situations

في كل واحد كلمه مستخدمه

غريبه هو معلافها في situation

تاني يبقي عشان تفهم ال situations

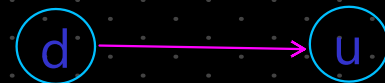
لازم تقرأهم بترتيب معين هو ببسألك

لو في حل واحد يعني شكل واحد لترتيب

ال situations اطبع 1 لو مفيش ولا حل

يعني كله معتمد علي كله اطبع 0 لو في

اكثر من حل اطبع 2



defined

used

انا مش هعرف ابدأ بحاجه غير باللي

ال In degree بتاعه بصفر .

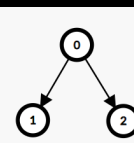
فانا ف اي وقت لو لقيت في نود ليها

ال In degree بصفر او انا خلصت

ال prerequisites فانا كذا جاهز اخذ

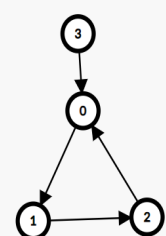
النود دي ولو قدامي اكثر من نود بنفس

الحاله دي فمممكن يكون في اكثر من حل



يعني مثلاً دي ليها كذا حل عشان

بعد ما اخلص ال 0 هيبقي قدامي 1 او 2



ودي ملهاش ولا حل عشان لما تخلص

ال 0 مش هبتلاقي حاجه

ال Indegree ليها 0

Following a sad and strange incident involving a room full of philosophers, several plates of spaghetti, and one too few forks, the faculty of the Department of Philosophy at ... University have been going through the papers of a recently deceased colleague. The faculty members were amazed to find numerous unpublished essays. They believe that the essays, collected into one volume, may constitute a major work of scholarship that will give their department some much-needed positive publicity. Naturally, all of the faculty members began to vie for the honor (to say nothing of the fame) of serving as editor of the collection.



After much debate, the faculty members have narrowed the list to two candidates. Both applicants were asked to explain how they would arrange the essays within the final book. Both have noted that many of the essays define terminology and concepts that are explored in other essays, and both have agreed to the basic principle that an essay that uses a term must itself define that term or appear after the essay that defines it.

One of the candidates has presented what he claims is the only possible arrangement of the essays under those constraints, and is arguing that he should be given the job simply because he has already done this major part of the work. The second candidate scoffs at this claim, insisting that there are many possible arrangements of the essays, and that an editor of true skill (himself) is needed to choose the optimal arrangement.

Write a program to determine if zero, one, or more than one arrangement of the essays is possible.

Input

There will be multiple test cases in the input.

Each test case will begin with a line with two integers, n ($1 \leq n \leq 1,000$) and m ($1 \leq m \leq 50,000$), where n is the number of essays, and m is the number of relationships between essays caused by sharing terms.

On each of the next m lines will be two integers, d and u ($1 \leq u, d \leq n, d \neq u$) which indicate that a term is defined in essay d and used in essay u .

The input will end with two 0's on their own line.

Output

For each test case, print a single line of output containing a '0' if no arrangement is possible, a '1' if exactly one arrangement is possible, or a '2' if multiple arrangements are possible (the output will be '2' no matter how many arrangements there are).

Sample Input

```
5 4
1 5
5 2
3 2
4 3
5 4
3 1
4 2
1 5
5 4
2 2
1 2
2 1
0 0
```

Sample Output

```
2
1
0
```

فانا هعمل q هحط فيه فالاول كل ال indegree بتاعهم ب 0 وبعدها اعدي علي الكيو واشوف كل نود انا حطيتها فيهم اقلل ال neighbors بتعها بواحد ولو في حاجه ال $\text{indegree} = 0$ احطها في الكيو ولما اخلص شغل علي النود دي امسحها من الكيو واحط اللي بعدها لو عدا علي كل النودز بيقا اكيد في 1 ولا اكتر هنعرف لكن لو معداش علي كل النودز يبقى مفيش حل طب هعرف مين هو حل واحد ولا اكتر هشوف لو في مره لقيت ال size بتاع الكيو اكبر من واحد يبقى في اكتر من حل

```
const int N = 1e+5, No_sol = 0, One_sol = 1, More_sols = 2;
int n, m, d, u;
```

```
vector<int> adj [N];
int IN [N];
bool vis[N];
```

```
int BFS() {
    int cnt = 0 ; // to count the number of visited nodes
    bool haveoptions = false ;
    queue<int > q;
    for (int i = 1; i ≤ n; ++i ) {
        if(IN[i] == 0) {
            q.push(i);
            vis[i] = true ;
        }
    }
}
```

```
while (!q.empty()) {
    if(q.size() > 1) haveoptions = true ;
    ++cnt ;
    int u = q.front() ;
    q.pop();
    for ( int v : adj[u]) {
        --IN[v];
        if (IN[v] == 0 && ! vis[v]) {
            q.push(v);
            vis[v] = true;
        }
    }
}
```

```
if(cnt ≠ n)
    return No_sol;
else if(haveoptions)
    return More_sols;
else
    return One_sol;
```

```
}
```

وشغلت ال visited array عشان
مشتغيش علي نود مرتين

```
void reset() {
    for (int i=1 ; i≤n ; ++i) adj[i].clear();
    memset(vis , Val: false, Size: sizeof vis);
    memset(IN , Val: 0, Size: sizeof IN );
}
```

```
int main() {
    while(( cin >> n >> m )&& n || m){
        reset();
        for (int i = 0; i < m; ++i) {
            cin >> d>>u;
            adj[d] .push_back(u);
            ++IN[u];
        }
        cout<<BFS()<<endl;
    }
}
```

انا هنا عملت reset function عشان
تفزي ال global variables عشان
هو هيسألني كذا مره

في عندي كذا time optimaization : اول حاجه اني بعمل كل مره memset افضي بيها ال visited في فكره احلي اني مفضيهاش بس اشوف كل نود هو انا زورتها في انهي تيست كيست يبقي بدل مخليها bool اخليها int وجواها رقم التيست كيست وكل reset هزود رقم التيست كيست بواحد بس

```
int vis[N];
int testcase;

int BFS() {
    int cnt = 0 ; // to count the number of visited nodes
    bool haveoptions = false ;
    queue<int> q;
    for (int i = 1; i ≤ n; ++i) {
        if(IN[i] == 0) {
            q.push(i);
            vis[i] = testcase ;
        }
    }

    while (!q.empty()) {
        if(q.size() > 1) haveoptions = true ;
        ++cnt ;
        int u = q.front() ;
        q.pop();
        for ( int v : adj[u]) {
            --IN[v];
            if (IN[v] == 0 && vis[v] != testcase) {
                q.push(v);
                vis[v] = testcase;
            }
        }
    }

    if(cnt != n)
        return No_sol;
    else if(haveoptions)
        return More_sols;
    else
        return One_sol;
}

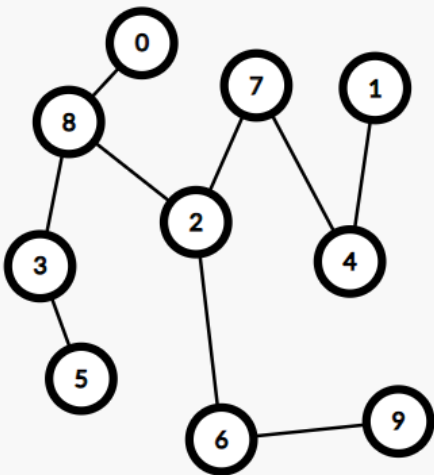
void reset() {
    for (int i=1 ; i≤n ; ++i) adj[i].clear();
    ++testcase; //time optimaizations
    memset(IN , Val: 0 , Size: sizeof IN );
}
```

انا بعمل ال reset قبل كل testcase يعني اول تيست كيست ب 1 مش 0

تاني optimaization اني مش محتاج ال visited array اصلا فحاجه ولو مسحتها الكود مش هيتأثر نهائي

Q: هيديك tree وعايزيك تجيب طول اطول path فيها يعني هنا اطول path هو من 1 ل 5

Sol: انا هبدأ من عند اي نقطه واروح اجيب ابعد نقطه عنها وبعدين اخذ النقطه اللي جبتها دي اجيب ابعد نقطه عنها كدا يبقا انا جيبت اطول path دايمًا بيسموا ال longest path دا ال tree diameter





```
const int N = 1e+5;
int n, m ,u, v;

vector<int> adj [N];
bool vis [N] ;

int farthrest_node;
int longestpath_length;

void DFS(int u ,int depth) {
    vis[u] = true;
    for (int v : adj[u]) {
        if(! vis[v]){
            DFS(u, v, depth+1);
        }
    }
    if(depth > longestpath_length){
        longestpath_length = depth ;
        farthrest_node = u ;
    }
}

int main() {
    cin >> n ;
    m= n-1;
    for (int i = 0; i < m; ++i) {
        cin >> u>>v;
        adj[u] .push_back(v);
        adj[v] .push_back(u);
    }
    DFS( u: 1, depth: 0);

    memset(vis, Val: false, Size: sizeof vis) ;
    longestpath_length = 0;

    DFS( u: farthrest_node, depth: 0);

    cout<<longestpath_length;
}
```

طلاما هبدا لحد ما اوصل للاخر بيقا كذا عايز .dfs

هبدا في الاول من عند اي نود واجيب ابعد نود من عندها

كل ما انزل في ال RECURSION هزود المتغير اللي

عامله برا بواحد اللي بيحيلي طول ال path ولما اجيب

اطول path ولما اجيب اطول path اخزن فيه النود دي بقا



هنا هو مدائش ال m اللي هي عدد ال edges

لكن اداني عدد النودز وقالوا انها tree

وانا عارف ان في ال h tree ال $m = n - 1$