Description | Hints | Submissions | Discussions | Notes

Round Trip

[illegible]	1 sec	512000KB	100
Difficulty	Time Limit	Memory	Score

80/80 XP | 30/30

Description

Zenithland has *n* cities and *m* roads between them. Your task is to check for the existence of a round trip that begins in a city, goes through two or more other cities, and finally returns to the starting city. Every intermediate city on the route has to be distinct.

Input Format

The first input line has two integers *n* and *m*: the number of cities and roads. The cities are numbered 1, 2, ..., *n*.

Then, there are *m* lines describing the roads. Each line has two integers *a* and *b*: there is a road between those cities.

Every road is between two different cities, and there is at most one road between any two cities.

Output Format

Print 'YES' if such round trip exists, otherwise print 'NO'.

Constraints

$1 \le n \le 10^5$

$1 \le m \le 2 \times 10^5$

$1 \le a, b \le n$

Sample Input 1 Copy

```
5 6
1 3
1 2
5 3
1 5
2 4
4 5
```

C++14 | 00:00:00 | 12 px

```
#define endl "\n"
using lli=long long int;
int n,m;
bool is_cycle=false;
vector<vector<int>>g;
vector<int>state;
vector<int>parent;
void dfs(int node,int par){
    state[node]=2;
    parent[node]=par;
    for(auto v:g[node]){
        if(v==parent[node]){
            continue;
        }
        if(state[v]==1){
            dfs(v,node);
        }
        else if(state[v]==2){
            is_cycle=true;
        }
    }
    state[node]=3;
}
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    cin>>n>>m;
    g.resize(n+1);
    state.assign(n+1,1);
    parent.resize(n+1);
    for(int i=0;i<m;i++){
        int a,b;
        cin>>a>>b;
        g[a].push_back(b);
        g[b].push_back(a);
```

Sample Tests | Manual Tests

Console | Run on Sample