Description | Hints | Submissions | Discussions | Notes

# Creating Teams

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

80/80 XP | 30/30

## Description

There are *n* students in AlgoZenith Course and *m* friendships between them. Your task is to divide the students into two teams in such a way that no two students in a team are friends. You can freely choose the sizes of the teams. The size of each team should be positive.

## Input Format

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

Then, there are *m* lines describing friendships. Each line has two integers *a* and *b*: students *a* and *b* are friends.

Every friendship is between two different students. You can assume that there is at most one friendship between any two students.

## Output Format

Print 'YES' if it's possible to divide students in two teams, 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 3
1 2
1 3
4 5
```

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

```
            dfs(v);
        }
        else if(color[v]!=col){
            is_valid=false;
        }
    }
    col=1^2^col;
}
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    cin>>n>>m;
    g.resize(n+1);
    color.assign(n+1,0);
    for(int i=0;i<m;i++){
        int a,b;
        cin>>a>>b;
        g[a].emplace_back(b);
        g[b].emplace_back(a);
    }
    col=1;
    for(int i=1;i<=n;i++){
        if(color[i]==0){
            dfs(i);
        }
    }
    if(is_valid){
        cout<<"YES"<<endl;
    }
    else{
        cout<<"NO"<<endl;
    }
    return 0;
}
```

Sample Tests | Manual Tests

Console | Run on Sample