

## ソーシャルネットワーク (Social Engineering)

問題名	ソーシャルネットワーク
入力	(インタラクティブ問題)
出力	(インタラクティブ問題)
実行時間制限	5秒
メモリ制限	256MB

あるソーシャルネットワークは  $n$  頂点  $m$  辺の無向グラフで表される。ここで各頂点は人であり、人と人の間に辺がある場合に限り、その 2 人は友達である。

このソーシャルネットワークのメンバーである Maria は、人を指名していくゲームを行うことにした。このゲームでは、まず Maria が友達を 1 人選んで指名する。その後、指名された人がその人の友達を 1 人選んで指名すること、を繰り返す。

ただし、同じ人が二度指名されることはあり得るが、どの 2 人のペアも一度しか指名に参加することはできない（つまり、一度人 A が人 B を指名した場合、今後人 A が人 B を指名したり、人 B が人 A を指名したりすることはできない）。言い換えると、指名の流れはグラフ上における、同じ辺を二度通らないウォーク（パス）で表される。

そして、もしある人のターンで、どの友達にも指名することが不可能になってしまった場合、この人は敗北する。

このゲームは常に Maria から始まり、面白いことに彼女はほとんど敗北したことがない。そこで残りの  $n-1$  人は互いに共闘し、Maria を敗北させようと考えた。あなたの課題は共闘を助けることである。

# 実装の詳細

あなたは以下の関数を実装しなければならない。

```
void SocialEngineering(int n, int m, vector<pair<int,int>> edges);
```

この関数は、頂点数  $n$ 、辺数  $m$  のグラフに対してゲームを行うものであり、一度しか呼び出されない。edges には  $m$  個の「2つの整数の組  $(u, v)$ 」からなり、頂点  $u$  と頂点  $v$  を結ぶ辺があることを意味する。頂点には1から  $n$  までの番号が付けられており、Maria の頂点番号は1である。さて、関数内では以下の2種類の関数を呼び出すことができる。

```
int GetMove();
```

この関数は、ゲーム開始時など、Maria のターンが行われるときに呼び出さなければならない。もし Maria のターンでないときに呼び出された場合、Wrong Answer となってしまう。この関数は以下のいずれかの値を返す。

- 整数  $v (2 \leq v \leq n)$ 。これは、Maria が人  $v$  を指名することを意味する。合法手ではない  $v$  の値になることはない。
- 0。これは、Maria が降参することを意味する。合法手が存在しなくなったとき、Maria は絶対に降参する。そうなった場合は関数 `SocialEngineering` を return しなければならない。すると Accepted の判定が得られる。

```
void MakeMove(int v);
```

この関数は、Maria 以外のターンが行われるときに呼び出さなければならない。  $v$  は次に指名する友達の番号を意味する。もし合法手ではなかったり、Maria のターンでこの関数が呼び出されたりした場合、Wrong Answer となってしまう。

最後に、もし Maria に勝ち筋が存在する場合、あなたのプログラムは**最初の `GetMove()` の呼び出しの前に `SocialEngineering` を return しなければならない**。そうすれば Accepted となる。

## 制約

- $2 \leq n \leq 2 \cdot 10^5$ .
- $1 \leq m \leq 4 \cdot 10^5$ .
- グラフは連結であり、多重辺（同じ2頂点間を結ぶ複数の辺）や自己ループ（同じ頂点を結んでいる辺）は存在しない。

## 小課題

もし Maria の勝ち筋が存在する場合、彼女は勝つために最適な動きをする。そうでない場合、彼女はあなたのミスを誘発させるために様々な戦略をとる。小課題 3 以外では、彼女は合法手が存在しなくなってしまった場合に限り、降参する。

1. (15 点)  $n, m \leq 10$ .
2. (15 点) Maria を除く全員は、高々 2 人としか友達ではない。
3. (20 点) もし Maria の勝ち筋が存在しない場合、彼女はすぐ降参する。
4. (25 点)  $n, m \leq 100$ .
5. (25 点) 追加の制約はない。

## 例

あなた	ジャッジシステム	説明
-	<code>SocialEngineering(5, 6, {{1,4}, {1,5}, {2,4}, {2,5}, {2,3}, {3,5}})</code>	関数 <code>SocialEngineering()</code> が、頂点数 5, 辺数 6 のグラフを引数として呼び出される。
<code>GetMove()</code>	Returns 4	Maria は人 4 を指名する。
<code>MakeMove(2)</code>	-	人 4 は人 2 を指名する。
<code>MakeMove(5)</code>	-	人 2 は人 5 を指名する。
<code>MakeMove(1)</code>	-	人 5 は Maria を指名する。
<code>GetMove()</code>	Returns 0	Maria には合法手が存在しないため、降参する。
Returns	-	あなたはこのゲームに勝利したため、関数 <code>SocialEngineering()</code> を return しなければならない。

あなた	ジャッジシステム	説明
-	<code>SocialEngineering(2, 1, {{1,2}})</code>	関数 <code>SocialEngineering()</code> が、頂点数 2, 辺数 1 のグラフを引数として呼び出される。
Returns	-	Maria にはこのグラフに対する勝ち筋があるため、あなたは <code>GetMove()</code> を呼び出す前に return することで降参しなければならない。

## 採点プログラムのサンプル (Sample Grader)

配布された `SocialEngineering.zip` の中に入っている、採点プログラムのサンプル `grader.cpp` は、入力を以下の形式で標準入力から受け取る。

- 1行目には、頂点数  $n$  と辺数  $m$  が書かれている。
- 続く  $m$  行には、各行ごとに2つの整数  $u, v$  が書かれている。これは頂点  $u$  と頂点  $v$  の間に辺があることを意味する。

採点プログラムのサンプルは入力を受け取った後、あなたの解答プログラムの関数 `SocialEngineering()` を呼び出す。ここで、採点プログラムのサンプルは、Maria の必勝法を実装していない、ただの例であることに注意すること。

採点プログラムのサンプルと、あなたの解答プログラムをコンパイルする方法の一つとして、Terminal 上で以下のコマンドを打つという手がある。

```
g++ -std=gnu++11 -O2 -o solution grader.cpp solution.cpp
```

ここで `solution.cpp` は、CMS 上で提出されるべき解答プログラムである。また、配布されたファイルの中に入っている入出力例を用いてプログラムを実行する方法の一つとして、Terminal 上で以下のコマンドを打つという手がある。

```
./solution < input.txt
```