SocialEngineering

Feladatnév	Social Engineering
Bemenet	Interaktív feladat
Kimenet	Interaktív feladat
Időkorlát	5 másodperc
Memóriakorlát	256 MB

A társadalmi háló egy *n* csúcsból és *m* élből álló, irányítatlan összefüggő gráf, ahol minden csúcs egy személyt jelöl. Két személyről akkor mondjuk, hogy barátok, ha van köztük él. Maria tagja egy társadalmi hálózatnak. Szereti kihívni a barátait különböző dolgokra. Ez azt jelenti, hogy először elvégez valamilyen egyszerű feladatot, majd kihívja az egyik barátját, hogy tegye meg ugyanazt. Ez a barát kihívja egy barátját, aki egy újabb barátját, és így tovább. Előfordulhat, hogy ugyanazt a személyt többször is kihívják, de minden egyes *baráti pár* legfeljebb egyszer vehet részt a kihívásban. (Ha *A* személy kihívja *B* személyt, akkor sem *A* személy, sem *B* személy nem hívhatja ki újra a másikat.) Más szóval a kihívások sorozata egy séta lesz a gráfban, amely soha nem használja többször ugyanazt az élet. Egy személy elveszíti a kihívást, ha ő következik a sorban és már nem tudja kihívni egyik barátját sem. A kihívásokat mindig Maria kezdi, és érdekesség, hogy csak néha veszít. A többiek, a fennmaradó *n*-1 ember úgy döntött, hogy együttműködnek annak érdekében, hogy Maria elveszítse a következő kihívást és a Te feladatod, hogy segíts nekik ezt véghezvinni.

Megvalósítás

A `void SocialEngineering(int n, int m, vector<pair<int,int>> edges);` eljárást kell megírnod, ami az *n* csúcsú és *m* élű gráfon játssza el a kihívást. Ezt az eljárást az értékelő egyszer fogja meghívni. Az élek listája pontosan *m* darab, egész számokból álló (*u*, *v*) számpárt tartalmaz. Az (*u*, *v*) számpár azt jelenti, hogy létezik él az *u* és a *v* csúcsok között. A csúcsok *1*-től *n*-ig vannak számozva, Maria mindig az *1* sorszámú. Az eljárásod meghívhatja a következő függvényeket:

```
int GetMove();
```

Ez a függvény mindannyiszor meghívódik, amikor Maria következik, mint például a játék elején. Ha akkor hívod meg, amikor nem Maria következik, akkor a `Wrong Answer` választ kapod. A függvény lehetséges visszatérési értékei:

- egy v egész szám, ahol $2 \leq v \leq n$. Ez azt jelenti, hogy Maria kihívja a v személyt. Ez mindig egy legális lépés.
- 0, ha Maria feladja a játékot. Maria mindig feladja a játékot, ha nincs legális lépése. Amikor ez megtörténik, a programodnak be kell fejeznie a `SocialEngineering()` eljárást, és visszakapod az `Accepted` értéket.

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

Ezt az eljárást mindannyiszor meg kell meghívni, amikor nem Maria következik. Azt jelenti, hogy az épp soron következő személy kihívja a v személyt. Ha ez nem legális lépés, vagy ha a hívás pillanatában Maria van soron, akkor `Wrong Answer` a válasz. Ha Maria a játék kezdetén nyerő stratégiával rendelkezik, a `SocialEngineering()` eljárásodnak az első `GetMove()` hívás *előtt* be kell fejeződnie. Ekkor az `Accepted` értéket kapod vissza.

Megszorítások

- $2 \leq n \leq 2 \cdot 10^5$.
- $1 \leq m \leq 4 \cdot 10^5$.
- A gráf összefüggő. A csúcsok minden nemrendezett párja legfeljebb egyszer fordul elő az élek közt, és minden él két különböző csúcs közt van, azaz egyszerű a gráf és nincs benne hurokél.

Részfeladatok

Maria mindig tökéletesen játszik abban az értelemben, hogy amikor létezik nyerő stratégiája, akkor mindig aszerint játszik. Ha nincs nyerő stratégiája, akkor megpróbálja a programodat hibázásra késztetni, különböző okos módszerekkel. Csak akkor fogja feladni a játékot, ha nincs legális lépése, kivéve a 3. részfeladatot.

1. részfeladat (15 pont) $n, m \leq 10$.
2. részfeladat (15 pont) Maria kivételével mindenkinek legfeljebb 2 barátja van.
3. részfeladat (20 pont) Maria azonnal feladja a játékot, kivéve, ha van nyerő stratégiája.
4. részfeladat (25 pont) $n, m \leq 100$.
5. részfeladat (25 pont) Nincs további megkötés.

Minta interakció

A programod lépése	Értékelő lépése	Magyarázat
-	`SocialEngineering(5, 6, {{1,4}, {1,5}, {2,4}, {2,5}, {2,3}, {3,5}})`	`SocialEngineering()` hívása 5 csúcsú és 6 élű gráfra.
`GetMove()`	`Returns 4`	Maria kihívja a 4. személyt.
`MakeMove(2)`	-	A 4. személy kihívja 2. személyt.
`MakeMove(5)`	-	A 2. személy kihívja az 5. személyt.
`MakeMove(1)`	-	Az 5. személy kihívja Mariat.
`GetMove()`	`Returns 0`	Marianak nincs legális lépése, így feladja a játékot.
`Returns`	-	A programod megnyerte a játékot és a `SocialEngineering()` eljárás befejeződik.

A programod lépése	Értékelő lépése	Magyarázat
-	`SocialEngineering(2, 1, {{1,2}})`	`SocialEngineering` hívása 2 csúcsú és 1 élű gráfra.
`Returns`	-	Marianak van nyerő stratégiája ezen a gráfon, így a versenyzőnek rögtön fel kell adnia a játékot.

Mintaértékelő

A `grader.cpp` mintaértékelő a feladat `SocialEngineering.zip` mellékletében van. A standard inputról olvas a következő formában:

- Az első sor a gráf csúcsainak *n* és éleinek *m* számát tartalmazza.
- Az ezt követő *m* sor az éleket tartalmazza, csúcspár formában.

A mintaértékelő beolvassa a bemenetet és meghívja a `SocialEngineering()` eljárást. Megjegyzés: a mintaértékelő nem tartalmazza Maria győztes stratégiáját és csak a mintainterakciók végrehajtására alkalmas. A mintaértékelő megoldásoddal együtt való fordításához parancssorból használhatod a `g++ -std=gnu++11 -O2 -o solution grader.cpp solution.cpp`
parancsot, ahol a `solution.cpp` a Te megoldásod, amit majd a CMS-be töltesz fel. A mellékletben

szereplő mintabemenettel futtathatod a programod, ha a parancssorba a `./solution < input.txt` utasítást írod.