

Социално инженерство

Име на задачата	Социално инженерство
Вход	Интерактивна задача
Изход	Интерактивна задача
Ограничение по време	5 секунди
Ограничение по памет	256 MB

Социалната мрежа се представя с ненасочен свързан граф с n върха и m ребра, в който всеки връх е човек и двама души са приятели, ако има ребро между тях.

Мария е член на тази социална мрежа. Тя обича да предизвиква приятелите си да правят различни неща. Това означава, че тя първо изпълнява някоя проста задача и след това предизвиква един от приятелите си да направи същото. Това предизвикателство след това ще пътува из мрежата. Може да се случи, че един и същи човек бива предизвикан повече от веднъж, но всяка неподредена двойка приятели могат да участват в предизвикателството най-много веднъж. (Веднъж щом човек A предизвика човек B , то нито човек A , нито човек B могат да предизвикат другия отново.) С други думи, предизвикателството ще бъде път в графа, който никога не използва ребро повече от един път. Човек губи предизвикателството, ако е негов ред и не може да предизвика някой от неговите приятели. Предизвикателствата винаги започват от Мария, и интересното е, че досега тя не е губила предизвикателства. Сега останалите $n - 1$ души са решили да си сътрудничат, за да направят, така че Мария да загуби следващото предизвикателство и вашата задача е да ги координирате.

Детайли по имплементацията

Трябва да имплементирате следната функция:

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

която играе играта за граф с n върха и m ребра. Тази функция ще бъде извикана веднъж от грейдъра. Списъкът `edges` ще съдържа точно m двойки цели числа (u, v) , което означава, че има ребро между върха u и върха v . Върховете са номерирани от 1 до n . Мария винаги е връх 1. Вашата функция може да извършва извиквания на следните функции: `int GetMove()` ;

Тази функция трябва да се извиква винаги, когато е ред на Мария, например в началото на

играта. Ако извикате тази функция, когато не е ред на Мария, ще получите `Wrong Answer`. Функцията може да върне една от следните стойности:

- едно цяло v , където $2 \leq v \leq n$. Това означава, че Мария предизвиква лицето с номер v . Това винаги ще бъде правилен ход.
- 0, ако Мария се откаже от играта. Мария винаги ще се откаже, ако няма валидни ходове. Когато това се случи, вашата програма трябва да приключи функцията `SocialEngineering`, и вие ще получите съобщение `Accepted`.

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

Тази функция трябва да се извиква винаги, когато не е ред на Мария. Това означава, че човекът, който е на ход, предизвиква човека с номер v . Ако това не е правилен ход или ако е ред на Мария в момента на извикването, тогава вие ще получите съобщение `Wrong Answer`. Ако Мария има печеливша стратегия в началото на играта, вашата програма трябва да прекрати функцията `SocialEngineering`, като върне още *преди* първото извикване на `GetMove`. След това ще получите съобщение `Accepted`.

Ограничения

- $2 \leq n \leq 2 \cdot 10^5$.
- $1 \leq m \leq 4 \cdot 10^5$.
- Графът е свързан. Всяка ненаредена двойка върхове ще се появи най-много веднъж като ребро и всяко ребро ще свързва два различни върхове.

Подзадачи

Мария винаги ще играе перфектно в смисъл, че ще прави печеливш ход винаги, когато има печеливша стратегия. Ако тя няма печеливша стратегия, тогава тя ще се опита да примами вашата програма да направи грешка, по различни хитри начини. Тя ще се откаже само ако няма валидни ходове, с изключение на подзадача 3.

1. (15 точки) $n, m \leq 10$.
2. (15 точки) Всеки, освен Мария, има най-много 2 приятеля.
3. (20 точки) Мария веднага ще се откаже освен ако няма печеливша стратегия.
4. (25 точки) $n, m \leq 100$.
5. (25 точки) Без допълнителни ограничения.

Примерно взаимодействие

Действие на състезателя	Действие на грейдъра	Обяснение
-	<code>SocialEngineering(5, 6, {{1,4}, {1,5}, {2,4}, {2,5}, {2,3}, {3,5}})</code>	<code>SocialEngineering</code> се извиква за граф с 5 върха и 6 ребра.
<code>GetMove</code>	Returns 4	Мария предизвиква човека с номер 4.
<code>MakeMove(2)</code>	-	Човек с номер 4 предизвиква човек с номер 2.
<code>MakeMove(5)</code>	-	Човек с номер 2 предизвиква човек с номер 5.
<code>MakeMove(1)</code>	-	Човек с номер 2 предизвиква Мария.
<code>GetMove</code>	Returns 0	Мария няма правилни ходове и затова се отказва.
Returns	-	Състезателят е спечелил играта и трябва да прекрати <code>SocialEngineering</code> , като се извика <code>return</code> .

Действие на състезателя	Действие на грейдъра	Обяснение
-	<code>SocialEngineering(2, 1, {{1,2}})</code>	<code>SocialEngineering</code> се извиква за граф с 2 върха и 1 ребро.
Returns	-	Мария има печеливша стратегия за този граф, така че състезателят трябва да прекрати функцията незабавно, за да се откаже.

Примерен грейдър

Предоставеният примерен градер, `grader.cpp`, в прикачения файл към задачата `SocialEngineering.zip`, чете входа от стандартния вход в следния формат:

- Първият ред съдържа броя на върховете, n , и броя на ребрата, m в графа.
- Следващите m реда съдържа по две цели числа u и v , които задават ребро между u и v .

Примерният грейдър чете входа и извиква `SocialEngineering` функцията от вашето решение. Обърнете внимание, че примерният грейдър не прилага печелившата стратегия

на Мария и е предоставен само да се изпробва взаимодействието.

За да компилирате примерния грейдър с вашето решение, можете да използвате следната команда в терминала: `g++ -std=gnu++11 -O2 -o solution grader.cpp solution.cpp` където `solution.cpp` е вашият файл с решение, който трябва да бъде изпратен в системата CMS. За да стартирате програмата с примерния вход, предоставен в прикачения файл, въведете следната команда от терминала: `./solution < input.txt`