

## Društveno upravljanje

Naziv zadatka	Društveno upravljanje
Datoteka za unos	interaktivni zadatak
Datoteka za ispis	interaktivni zadatak
Vremensko ograničenje	5 sekundi
Memorijsko ograničenje	256 megabajta

Društvena mreža sastoji se od neusmjerenog povezanog grafa s  $n$  čvorova i  $m$  bridova, gdje je svaki čvor osoba i dvije osobe su prijatelji ako postoji brid između njih. Martina je članica društvene mreže. Voli zadavati prijateljima razne izazove. To znači da sama prvo napravi nešto opasno i onda izazove jednu od svojih prijateljica da učini to isto. Zatim ta prijateljica izazove neku od svojih prijateljica i tako dalje... Taj izazov onda putuje kroz društvenu mrežu. Može se dogoditi da se istu osobu izazove više no jednom, ali svaki neuređeni par prijatelja može sudjelovati u izazovu najviše jednom. (Jednom kada osoba  $A$  izazove osobu  $B$ , otada se osobe  $A$  i  $B$  ne mogu više međusobno izazvati) Drugim riječima, izazov će biti šetnja grafom koja ne prolazi nijednim bridom više no jedanput.

Osoba je izgubila izazov kada je na redu te ne može izazvati nijednu od svojih prijateljica. Izazove uvijek započinje Martina koja rijetko gubi. Preostalih  $n-1$  prijateljica odlučilo se urotiti protiv Martine kako bi izgubila sljedeći izazov te vaš je zadatak uskladiti ih.

## Implementacija

Vaš je zadatak implementirati sljedeću funkciju:

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

koja vrši zadani izazov na grafu od  $n$  čvorova i  $m$  bridova. Tu funkciju će jednom pozvati grader. Vektor `edges` sadržavat će točno  $m$  parova brojeva  $(u, v)$ , koji označavaju brid koji spaja čvorove  $u$  i  $v$ . Čvorovi su označeni brojevima od  $1$  do  $n$ . Martina je uvijek čvor  $1$ . Vaša funkcija smije pozvati sljedeće metode:

```
int GetMove();
```

Ova metoda trebala bi biti pozvana svaki put kad je Martina na potezu, primjerice na početku izazova. Ako pozovete ovu metodu kada Martina nije na redu, dobit ćete `Wrong Answer`. Ova metoda može vratiti jednu od sljedećih vrijednosti:

- broj  $v$ , gdje je  $2 \leq v \leq n$ . To znači da je sljedeća osoba koju Martina izaziva osoba  $v$ . To će uvijek biti dozvoljen potez.
- 0, ako se Martina predaje. Martina će se uvijek predati ako više nema dozvoljenih poteza. Kada se to dogodi, trebali biste dopustiti funkciji `SocialEngineering` da vrati, i dobit ćete `Accepted`.

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

Ova metoda trebala bi biti pozvana kadgod Martina nije na potezu. To znači da osoba koju će osoba koja je trenutno na redu izazvati je upravo osoba  $v$ . Ako to nije dozvoljen potez ili je trenutno Martinin potez, dobit ćete `Wrong Answer`.

Ako Martina ima pobjedničku strategiju na početku igre, trebali biste dopustiti funkciji `SocialEngineering` da vrati *prije* prvog poziva `GetMove()`. U tom slučaju dobit ćete `Accepted`.

## Ograničenja

- $2 \leq n \leq 2 \cdot 10^5$ .
- $1 \leq m \leq 4 \cdot 10^5$ .
- Graf je povezan. Svaki neuređeni par čvorova pojavljuje se kao brid najviše jednom te svaki brid spaja dva različita čvora.

## Subtasks

Martina će uvijek igrati savršeno tj. uvijek će odigrati pobjednički potez ako ima pobjedničku strategiju. Ako nema pobjedničku strategiju, pokušat će natjerati vaš program da napravi grešku na razne lukave načine. Predat će se samo kada nema dozvoljenih poteza, osim u Podzadatku 3.

1. (15 bodova)  $n, m \leq 10$ .
2. (15 bodova) Svi osim Martine imaju najviše dva prijatelja.
3. (20 bodova) Martina će se odmah predati ukoliko nema pobjedničku strategiju.
4. (25 bodova)  $n, m \leq 100$ .
5. (25 bodova) Nema dodatnih ograničenja.

## Primjer interakcije

Potez natjecatelja	Potez gradera	Objašnjenje

Potez natjecaljtea	Potez gradera	Objašnjenje
-	<code>SocialEngineering(5, 6, {{1,4}, {1,5}, {2,4}, {2,5}, {2,3}, {3,5}})</code>	<code>SocialEngineering</code> pozvana je s grafom od 5 čvorova i 6 bridova.
<code>GetMove()</code>	Vraća 4 4	Martina izaziva osobu 4.
<code>MakeMove(2)</code>	-	Osoba 4 izaziva osobu 2.
<code>MakeMove(5)</code>	-	Osoba 2 izaziva osobu 5.
<code>MakeMove(1)</code>	-	Osoba 5 izaziva Martinu.
<code>GetMove()</code>	Vraća 0	Martina nema dozvoljenih poteza te se predaje.
Vraća	-	Natjecatelj osvaja izazov te bi trebao dopustiti funkciji <code>SocialEngineering</code> da vrati.

Potez natjecaljtea	Potez gradera	Objašnjenje
-	<code>SocialEngineering(2, 1, {{1,2}})</code>	<code>SocialEngineering</code> se poziva s grafom od 2 čvora i 1 brida.
Vraća	-	Martina ima pobjedničku strategiju te bi funkcija odmah trebala vratiti kako bi se predala.

## Primjer gradera

Dani primjer gradera, `grader.cpp`, u prilogu zadatak `SocialEngineering.zip`, čita unos iz standardnog unosa na sljedeći način:

- Prvi red sadrži broj čvorova,  $n$ , i broj bridova,  $m$  u grafu.
- Sljedećih  $m$  redova sadrži bridove kao parove.

Primjer gradera čita unos i poziva funkciju `SocialEngineering` unutar natjecateljevog rješenja. Primjetite da grader ne implementira Martininu pobjedničku strategiju već samo daje primjer interakcije. Kako biste kompajlirali sample grader zajedno sa svojim rješenjem, koristite sljedeću naredbu u terminalu:

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

gdje je `solution.cpp` datoteka vašeg rješenja koje trebate i poslati na CMS. Kako biste pokrenuli

svoje rješenje s primjerom danom u prilogu, upišite sljedeću naredbu u terminal:

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