

SocialEngineering

Tehtävän nimi	Social Engineering
Syötetiedosto	Interaktiivinen tehtävä
Tulostetiedosto	Interaktiivinen tehtävä
Aikaraja	5 sekuntia
Muistiraja	256 megabytes

Sosiaalinen verkosto koostuu suuntaamattomasta (undirected) verkosta, jossa on n solmua ja m kaarta, missä jokainen solmu on yksi henkilö, ja jos kaksi henkilöä ovat ystäviä keskenään, heidän välillään on kaari. Maria on tämän sosiaalisen verkoston jäsen. Hän tykkää haastaa ystäviään erilaisten asioiden tekemiseen. Se tarkoittaa sitä, että hän ensin tekee jonkin asian, ja sen jälkeen haastaa yhden ystävistään tekemään saman asian. Tämä ystävä sitten haastaa yhden ystävänsä, joka haastaa yhden ystävistään, ja niin edelleen. Sama henkilö voi saada haasteen enemmän kuin yhden kerran, mutta jokainen järjestämätön (unordered) pari henkilöitä voi ottaa osaa haasteeseen vain kerran. (Kun henkilö A haastaa henkilön B , sen jälkeen niin A kuin B :kään ei saa haastaa toinen toistaan uudestaan). Toisin sanoen, haaste kulkee verkossa siten ettei se koskaan saavuta samaa kaarta uudestaan. Henkilö häviää haasteen jos on hänen vuoronsa, eikä hän voi enää haastaa yhtäkään ystävänsä. Haasteet aloitetaan aina Mariasta, ja hän häviää harvoin. Nyt jäljellä olevat $n-1$ ihmistä ovat päättäneet tehdä yhteistyötä saadakseen Marian häviämään seuraavan haasteen, ja on sinun tehtäväsi koordinoita tätä yritystä.

Toteutus

Sinun täytyy implementoida funktio:

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

joka pelaa peliä verkossa jossa on n solmua ja m kaarta. Arvioija (grader) kutsuu tätä funktiota kerran. Lista `edges` sisältää tarkalleen m paria kokonaislukuja (u, v) , tarkoittaen että kaari kulkee solmujen u ja solmun v välillä. Solmut numeroidaan 1:stä n :ään. Maria on aina solmu 1. Sinun funktiosi voi tehdä kutsuja seuraaviin metodeihin:

```
int GetMove();
```

Tätä metodia pitää kutsua aina kun on Marian vuoro, kuten pelin alussa. Jos kutsut tätä metodia silloin kun ei ole Marian vuoro, saat vastauksen `Wrong answer`. Metodi voi palauttaa yhden seuraavista arvoista:

- kokonaisluku v , missä $2 \leq v \leq n$. Tämä tarkoittaa että Maria haastaa henkilön indeksillä v . Tämä on aina sallittu siirto.
- 0, jos Maria luovuttaa pelin. Maria luovuttaa aina, jos hänellä ei ole laillisia siirtoja. Kun tämä tapahtuu, funktiosi `SocialEngineering` tulee palautua, ja saat vastauksen `Accepted`.

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

Tätä metodia tulee kutsua aina kun ei ole Marian vuoro. Se tarkoittaa että vuorolla oleva henkilö haastaa henkilön v . Jos se ei ole sallittu siirto tai on Marian vuoro kun teet kutsun, tulet saamaan vastauksen `Wrong answer`.

Jos Marialla on voittava strategia pelin alussa, ohjelmasi pitäisi antaa funktion `SocialEngineering` palautua (return) *ennen* metodin `GetMove()` ensimmäistä kutsua. Tulet sitten saamaan vastauksen `Accepted`.

Rajoitteet

- $2 \leq n \leq 2 \cdot 10^5$.
- $1 \leq m \leq 4 \cdot 10^5$.
- Verkko on yhtenäinen. Jokaisen järjestämättömän (unordered) parin solmuja välillä voi olla kaari vain kerran, ja jokainen kaari kulkee kahden eri solmun välillä.

Alitehtävät

Maria pelaa aina täydellisesti siinä mielessä, että hän tekee voittavia siirtoja aina kun hänellä on voittava strategia. Jos hänellä ei ole voittavaa strategiaa, hän yrittää houkutella ohjelmasi tekemään virheen monenlaisin älykkäin keinoin. Hän luovuttaa vain jos hänellä ei ole sallittuja siirtoja, paitsi alitehtävässä 3.

1. (15 points) $n, m \leq 10$.
2. (15 points) Kaikilla paitsi Marialla on korkeintaan 2 ystävää.
3. (20 points) Maria luovuttaa välittömästi ellei hänellä ole voittavaa strategiaa.
4. (25 points) $n, m \leq 100$.
5. (25 points) Ei rajoitteita.

Esimerkkitoiminta

Sinun toimintasi	Arvioijan (grader) toiminta	Selitys
-	<code>SocialEngineering(5, 6, {{1,4}, {1,5}, {2,4}, {2,5}, {2,3}, {3,5}})</code>	<code>SocialEngineering</code> kutsutaan verkolla jossa on 5 solmua ja 6 kaarta
<code>GetMove()</code>	Palauttaa 4	Maria haastaa henkilön 4.
<code>MakeMove(2)</code>	-	Henkilö 4 haastaa henkilön 2.
<code>MakeMove(5)</code>	-	Henkilö 2 haastaa henkilön 5.
<code>MakeMove(1)</code>	-	Henkilö 5 haastaa Marian.
<code>GetMove()</code>	Palauttaa 0	Marialla ei ole sallittuja siirtoja, joten hän luovuttaa.
Palautuu	-	Olet voittanut pelin, ja funktion <code>SocialEngineering</code> tulee palautua

Sinun toimintasi	Arvioijan toiminta	Selitys
-	<code>SocialEngineering(2, 1, {{1,2}})</code>	<code>SocialEngineering</code> kutsutaan verkolle jossa on 2 solmua ja 1 kaari
Palautuu	-	Marialla on voittava strategia tässä verkossa, joten ohjelmasi tulee palautua tekemättä yhtään <code>GetMove()</code> kutsua.

Esimerkkiarvioija (grader)

Annettu esimerkkiarvioija, `grader.cpp`, tehtävän liitteessä `SocialEngineering.zip`, lukee syötteen standardisyötteestä seuraavassa muodossa:

- Ensimmäinen rivi sisältää solmujen määrän, n , ja kaarien määrän, m , verkossa.
- Seuraavat m riviä sisältävät jokainen kaksi kokonaislukua u ja v , tarkoittaen että niiden välillä on kaari.

Esimerkkiarvioija lukee syötteen ja kutsuu funktiota `SocialEngineering` ratkaisussa. Muista, että esimerkkiarvioija tarjoaa vain esimerkkivuorovaikutusta eikä pelaa Marian voittavalla strategialla. Kääntääksesi esimerkkiarvioijan ratkaisullasi, käytä seuraavaa komentoa komentorivillä: `g++ -std=gnu++11 -O2 -o solution grader.cpp solution.cpp` missä `solution.cpp` on

ratkaisutiedostosi submitoitavaksi CMS:ään. Ajaaksesi ohjelman esimerkkisyötteellä liitteessä, aja komentorivillä `./solution < input.txt`