

Socialinė inžinerija (SocialEngineering)

Užduoties pavadinimas	Socialinė inžinerija
Įvesties failas	Interaktyvi užduotis
Išvesties failas	Interaktyvi užduotis
Laiko ribojimas	5 sekundės
Atminties ribojimas	256 megabaitai

Socialinį tinklą sudaro nekryptinis jungus grafas su n viršūnių ir m briaunų, kur kiekviena viršūnė yra žmogus, ir jei du žmonės yra draugai, tai tarp jų yra kraštinė.

Marija yra šio tinklo narė. Ji mėgsta kviesti draugus išmėginti įvairius dalykus. Tai reiškia, kad ji pirmiausia padaro kokią nors paprastą užduotį, tada meta iššūkį vienam iš savo draugų atlikti tą patį dalyką. Šis draugas mes šį iššūkį vienam iš savo draugų, kuris mes šį iššūkį vienam iš savo draugų ir t.t. Gali nutikti taip, kad tas pats žmogus gaus iššūkį daugiau nei vieną kartą, bet kiekviena draugų pora, kur poros tvarka nesvarbi, gali dalyvauti šiame iššūkyje ne daugiau nei vieną kartą. (Kai žmogus A meta iššūkį žmogui B , tai nei žmogus A nei žmogus B nebegali vienas kitam mesti iššūkio.) Kitais žodžiais, iššūkiai bus judėjimas grafu, nenaudojant tos pačios briaunos daugiau nei kartą.

Žmogus pralaimi iššūkį, jei jam tenka eilė mesti iššūkį, bet jis nebegali mesti iššūkio nei vienam iš savo draugų. Šiuos iššūkius visada pradeda Marija, bet ji retai pralaimi. Dabar likę $n - 1$ žmonės nusprendė bendradarbiauti taip, kad Marija pralaimėtų kitą iššūkį. Jūsų darbas – sukoordinuoti šį bendradarbiavimą.

Implementacija

Jums reikia realizuoti funkciją:

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

kuri žaidžia žaidimą grafe su n viršūnių ir m kraštinių. Vertinimo programa iškvies šią funkciją vieną kartą. Sąrašė `edges` bus lygiai m sveikųjų skaičių porų (u, v) , kurios reiškia, kad yra briauna tarp viršūnės u ir viršūnės v . Viršūnės sunumeruotos nuo 1 iki n . Marija visada yra viršūnė 1. Jūsų funkcija gali kviesti šiuos metodus:

```
int GetMove();
```

Šis metodas turėtų būti iškviestas kai ateina Marijos eilė, pavyzdžiui, žaidimo pradžioje. Jei iškviessite šį metodą kai eilė ne Marijos, gausite verdiktą `Wrong Answer` (Neteisingas atsakymas). Šis metodas gali grąžinti vieną iš šių reikšmių:

- sveikąjį skaičių v ($2 \leq v \leq n$). Tai reiškia, kad Marija meta iššūkį žmogui su indeksu v . Šis ėjimas visada bus galimas.
- 0, jei Marija pralaimi. Marija visada pralaimės, jei neturės galimų ėjimų. Kai taip nutinka, jūsų programa turėtų leisti funkcijai `SocialEngineering` baigti darbą (angl. „return“) ir gausite verdiktą `Accepted` (Priimta).

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

Šis metodas turėtų būti iškviestas, kai eilė ne Marijai. Tai reiškia, kad žmogus, kuriam dabar eilė, meta iššūkį žmogui v . Jei šis ėjimas negalimas arba jei šiuo metu eilė Marijai, gausite verdiktą `Wrong Answer` (Neteisingas atsakymas).

Jei žaidimo pradžioje Marija turi laiminčiąją strategiją, jūsų programa turėtų leisti funkcijai `SocialEngineering` baigti darbą (angl. „return“) prieš pirmą kartą kviečiant `GetMove()`. Tokiu atveju gausite verdiktą `Accepted` (Priimta).

Ribojimai

- $2 \leq n \leq 2 \cdot 10^5$.
- $1 \leq m \leq 4 \cdot 10^5$.
- Grafas yra jungus. Kiekviena viršūnių pora (tvarka nesvarbi) bus pateikta kaip briauna ne daugiau nei vieną kartą. Kiekviena briauna jungs dvi skirtingas viršūnes.

Dalinės užduotys

Marija visada žais tobulai. T.y. ji visada rinksis laimintį ėjimą, jei turi laiminčiąją strategiją. Jei ji neturės laiminčiosios strategijos, bandys suklaidinti jūsų programą įvairiais protingais būdais. Ji pralaimės tik tada, jei neturės galimo ėjimo, išskyrus dalinę užduotį 3.

1. (15 taškų) $n, m \leq 10$.
2. (15 taškų) Visi (išskyrus Mariją) turės ne daugiau nei du draugus.
3. (20 taškų) Marija iš karto pralaimės, jei ji neturi laiminčiosios strategijos.
4. (25 taškai) $n, m \leq 100$.

5. (25 taškai) Jokių papildomų ribojimų.

Komunikacijos pavyzdys

Tavo veiksmas	Vertinimo programos veiksmas	Paaiškinimas
-	<code>SocialEngineering(5, 6, {{1,4}, {1,5}, {2,4}, {2,5}, {2,3}, {3,5}})</code>	<code>SocialEngineering</code> iškviečiamas grafui su 5 viršūnėmis ir 6 briaunomis.
<code>GetMove()</code>	Returns 4	Marija meta iššūkį žmogui 4.
<code>MakeMove(2)</code>	-	Žmogus 4 meta iššūkį 2.
<code>MakeMove(5)</code>	-	Žmogus 2 meta iššūkį 5.
<code>MakeMove(1)</code>	-	Žmogus 5 meta iššūkį Marijai.
<code>GetMove()</code>	Returns 0	Marija neturi galimų ėjimų, tad ji pralaimi.
Gražina (angl. „Returns“)	-	Tu laimėjai žaidimą, ir turėtum leisti funkcijai <code>SocialEngineering</code> baigti darbą (angl. „return“).

Tavo veiksmas	Vertinimo programos veiksmas	Paaiškinimas
-	<code>SocialEngineering(2, 1, {{1,2}})</code>	<code>SocialEngineering</code> iškviesta grafui su 2 viršūnėmis ir 1 briauna.
Gražina (angl. „Returns“)	-	Šiuo atveju Marija turi laiminčią strategiją, tad tu pralaimi ir turėtum iš karto leisti funkcijai baigti darbą (angl. „return“), nė karto nekviečiant <code>GetMove()</code> .

Pavyzdinė vertinimo programa

Pateikta pavyzdinė vertinimo programa `grader.cpp` užduoties sąlygos prisegtuke `SocialEngineering.zip`. Ji nuskaito įvestį iš standartinės įvesties tokiu formatu:

- Pirmoje eilutėje yra grafo viršūnių kiekis n ir briaunų kiekis m .
- Kitose m eilučių yra du sveikieji skaičiai u ir v , nurodantys, jog tarp u ir v yra briauna.

Pavyzdinė vertinimo programa nuskaito įvestį ir iškviečia `SocialEngineering` funkciją, pateiktą dalyvio sprendime. Atkreipkite dėmesį, kad pavyzdinė vertinimo programa nerealizuoja Marijos laiminčiosios strategijos. Ši programa pateikta tik pavyzdinei interakcijai.

Kad sukompiliuotumėte pavyzdinę vertinimo programą kartu su savo sprendimu, galite naudoti šią komandą komandinėje eilutėje (terminale): `g++ -std=gnu++11 -O2 -o solution grader.cpp solution.cpp`, kur `solution.cpp` yra jūsų sprendimų failas, kurį norite įkelti į CMS. Kad paleistumėte programą su pavyzdiniu failu (kuris pateiktas prisegtuke), paleiskite šią komandą terminale: `./solution < input.txt`