

Sosyal Mühendislik

Problem Adı	Sosyal Mühendislik
Girdi dosyası	Etkileşimli görev
Çıktı dosyası	Etkileşimli görev
Zaman limiti	5 saniye
Hafıza limiti	256 megabytes

Bir sosyal ağ n düğümlü ve m kenarlı bir yönsüz bağlı çizgeden oluşur. Bu çizgede her düğüm bir kişiyi temsil eder ve iki kişi arasında bir kenar var ise bu iki kişi arkadaşlardır. Maria bu sosyal ağın bir üyesidir. Maria arkadaşlarına meydan okumayı sever. Yani, Maria önce basit bir görevi yerine getirir ve arkadaşlarına aynı şeyi yapmaları konusunda meydan okur. Sonrasında bu meydan okuma oyunu arkadaştan arkadaşta giderek ağ boyunca yayılır. Bu oyun sürecinde bir kişiye birden fazla defa meydan okunabilir ancak sırasız bir arkadaş ikilisi arasında sadece bir kere meydan okuma gerçekleşebilir. (A kişisi B kişisine meydan okuduktan sonra ne A kişisi B kişisine ne de B kişisi A kişisine meydan okuyabilir.) Başka bir deyişle, bir meydan okuma zinciri çizgenin hiçbir kenarını bir kereden fazla kullanmayan bir gezintidir. Meydan okuma sırası kendine gelen ancak hiç bir arkadaşına meydan okuyamayan kişi meydan okuma oyununu kaybeder. Meydan okuma oyununu her zaman Maria başlatır, ve ilginçtir, Maria bu oyunu şimdiye kadar hiç kaybetmemiştir. Diğer $n - 1$ kişi Maria'nın bir sonraki meydan okuma oyununu kaybetmesi için beraberce hareket etmeye karar verirler ve sizin göreviniz onlara bu konuda yardımcı olmaktır.

Gerçekleştirme (Implementation)

Bu oyunu n düğümlü ve m kenarlı bir çizge üzerinde oynayan `void SocialEngineering(int n, int m, vector<pair<int,int>> edges);` fonksiyonunu gerçekleştireceksiniz. Bu fonksiyon değerlendirici tarafından bir kere çalıştırılacaktır. `edges` isimli liste herbiri bir kenara karşılık gelen m adet (u, v) tamsayı ikilisinden oluşmaktadır. Yani, bu listedeki bir (u, v) ikilisi u düğümü ve v düğümü arasında bir kenar olduğunu göstermektedir. Düğümler 1 'den n 'ye kadar tamsayılar ile numaralandırılmıştır. Maria her zaman 1 numaralı düğümdür. Fonksiyonunuz aşağıdaki fonksiyonlara çağrı yapabilir:

```
int GetMove();
```

Bu fonksiyon sıra Maria'ya geldiğinde çağırılmalıdır. Oyunun başında sıra Maria'dadır. Bu

fonksiyonu sıra Maria'da değilken çağırırsanız `Wrong Answer` (Yanlış Cevap) hükmünü alırsınız. Bu fonksiyon aşağıdaki değerlerden birini dönebilir:

- $2 \leq v \leq n$ olmak üzere bir v tamsayısı. Bu durumda Maria v numaralı arkadaşına meydan okumaktadır. bu meydan okuma kurallara uygun olacaktır.
- Maria yenildiğini kabul ederse 0 dönecektir. Maria kurallara uygun hamlesi kalmadığında yenildiğini kabul edecektir. Bu olduğunda `SocialEngineering()` fonksiyonunuz dönmelidir. bu durumda `Accepted` (Kabul Edildi) yargısını alacaksınız.

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

Bu fonksiyon sıra Maria'dan başka birinde ise çağırılmalıdır. Bu çağrı ile sıradaki kişi v numaralı kişiye meydan okumaktadır. Çağrı esnasında sıra Maria'da ise veya bu hamle kurallara uygun bir hamle değilse `Wrong Answer` yargısını alırsınız. Eğer oyun başında Maria'nın oyunu kazanmayı garanti edecek bir stratejisi var ise `SocialEngineering()` fonksiyonu `GetMove()` fonksiyonunu hiç çağırmadan dönmelidir. Bu durumda `Accepted` yargısını alırsınız.

Kısıtlar

- $2 \leq n \leq 2 \cdot 10^5$.
- $1 \leq m \leq 4 \cdot 10^5$.
- Çizge bağlıdır. Her sırasız düğüm ikilisi en fazla bir kere kenar olarak listelenebilir ve her kenar birbirinden farklı iki düğüm arasındadır.

Altgörevler

Maria oyunu olabilecek en iyi şekilde oynayacaktır. Kazanmasını garanti eden bir strateji var ise o stratejiyi oynayacaktır. Kazanmasını garanti eden bir stratejisi yoksa akıllı şekilde tuzaklar kurup sizin programınızı hataya zorlayacaktır. Altgörev 3 haricindeki bütün altgörevlerde, Maria sadece kurallara uygun hamlesi kalmadığında yenildiğini kabul edecektir.

1. (15 points) $n, m \leq 10$.
2. (15 points) Maria haricindeki kişilerin herbirinin en fazla 2 arkadaşı vardır.
3. (20 points) Maria kazanmasını garanti edecek bir stratejisi yoksa hemen yenildiğini kabul edecektir.
4. (25 points) $n, m \leq 100$.
5. (25 points) Başkaca kısıt yoktur.

Örnek Etkileşim

Sizin hamleleriniz	Değerlendiricinin hamleleri	Açıklama
-	<code>SocialEngineering(5, 6, {{1,4}, {1,5}, {2,4}, {2,5}, {2,3}, {3,5}})</code>	<code>SocialEngineering()</code> fonksiyonu 5 düğümlü ve 6 kenarlı bir çizge ile çağrılmaktadır.
<code>GetMove()</code>	Returns 4	Maria 4 numaralı kişiye meydan okumaktadır.
<code>MakeMove(2)</code>	-	4 numaralı kişi 2 numaralı kişiye meydan okumaktadır.
<code>MakeMove(5)</code>	-	2 numaralı kişi 5 numaralı kişiye meydan okumaktadır.
<code>MakeMove(1)</code>	-	5 numaralı kişi Maria'ya meydan okumaktadır.
<code>GetMove()</code>	Returns 0	Maria kurallara uygun bir hamlesi kalmadığı için oyunu kaybettiğini kabul etmektedir.
Returns	-	Oyunu kazandınız. <code>SocialEngineering()</code> fonksiyonunu döndürün.

Sizin hamleleriniz	Değerlendiricinin hamleleri	Açıklama
-	<code>SocialEngineering(2, 1, {{1,2}})</code>	<code>SocialEngineering</code> fonksiyonu 2 düğümlü ve 1 kenarlı bir çizge ile çağrılmaktadır.
Returns	-	Bu çizgede Maria'nın kazanmayı garanti eden bir stratejisi vardır. Bu yüzden siz hiç bir <code>GetMove()</code> çağrısı yapmadan dönmelisiniz.

Örnek Değerlendirici

Görev eklentisi `SocialEngineering.zip`'de verilen örnek değerlendirici `grader.cpp` girdiyi standart girdiden aşağıdaki formatta okumaktadır:

- İlk satırda çizgenin düğüm sayısı n ve kenar sayısı m vardır.
- Sonraki m satırın her biri 2 tamsayıdan (u ve v) oluşmaktadır. Bu satır u ve v düğümleri arasında bir kenar olduğunu göstermektedir.

Örnek değerlendirici girdiyi okur ve kullanıcının çözümündeki `SocialEngineering()` fonksiyonunu çağırır. Örnek değerlendirici Maria'nın kazanan stratejisini gerçeklememektedir, sadece örnek etkileşim için verilmiştir.

Örnek değerlendiriciyi çözümünüzle beraber derlemek için aşağıdaki komutu terminale yazabilirsiniz: `g++ -std=gnu++11 -O2 -o solution grader.cpp solution.cpp`

Burada `solution.cpp` CMS'e yükleyeceğiniz çözüm dosyanızdır. Programı eklentide verilen örnek girdi ile çalıştırmak için, aşağıdaki komutu terminale yazınız:

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