

## Κοινωνική Μηχανική (Social Engineering)

Όνομα Προβλήματος	Κοινωνική Μηχανική (Social Engineering)
Αρχείο εισόδου	Διαδραστικό πρόβλημα (Interactive task)
Αρχείο εξόδου	Διαδραστικό πρόβλημα (Interactive task)
Χρονικό όριο	5 δευτερόλεπτα
Όριο μνήμης	256 megabytes

Ένα κοινωνικό δίκτυο αποτελείται από έναν μη κατευθυνόμενο συνδεδεμένο γράφο με  $n$  κορυφές και  $m$  ακμές, όπου κάθε κορυφή είναι ένα άτομο, και δύο άτομα είναι φίλοι, εάν υπάρχει μια ακμή μεταξύ τους.

Η Μαρία είναι μέλος αυτού του κοινωνικού δικτύου. Της αρέσει να βάζει προκλήσεις στους φίλους της να κάνουν διάφορα πράγματα. Αυτό σημαίνει ότι αρχικά εκτελεί μια απλή ενέργεια και στη συνέχεια προκαλεί τους φίλους της να κάνουν το ίδιο. Αυτή η πρόκληση στη συνέχεια θα ταξιδέψει στο δίκτυο. Μπορεί να συμβεί το ίδιο άτομο να δεχθεί την πρόκληση περισσότερες από μια φορές, αλλά κάθε μη ταξινομημένο ζεύγος φίλων μπορεί να πάρει μέρος στην πρόκληση το πολύ μία φορά. (Από τη στιγμή που ένα άτομο  $A$  προκαλεί ένα άτομο  $B$ , τότε ούτε το άτομο  $A$  ούτε το άτομο  $B$  μπορούν να προκαλέσουν τον άλλον ξανά). Με άλλα λόγια, η πρόκληση θα είναι μια διάσχιση του γράφου, η οποία δεν χρησιμοποιεί ποτέ την ίδια ακμή παραπάνω από μία φορά.

Ένα άτομο χάνει την πρόκληση εάν είναι η σειρά του και δεν μπορεί να προκαλέσει κανέναν από τους φίλους του. Οι προκλήσεις ξεκινούν πάντα από την Μαρία, η οποία περιέργως δεν έχει χάσει καμία πρόκληση μέχρι τώρα. Τώρα τα απομείναντα  $n - 1$  άτομα αποφάσισαν να συνεργαστούν έτσι ώστε να κάνουν την Μαρία να χάσει την επόμενη πρόκληση, και αποτελεί δική σου δουλειά να συντονίσεις αυτή την προσπάθεια.

## Υλοποίηση

Πρέπει να υλοποιήσετε την συνάρτηση:

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

η οποία παίζει το παιχνίδι σε έναν γράφο με  $n$  κορυφές και  $m$  ακμές. Η συνάρτηση αυτή θα κληθεί μια φορά από τον βαθμολογητή. Η λίστα `edges` θα περιέχει ακριβώς  $m$  ζεύγη ακεραίων  $(u, v)$ , το οποίο σημαίνει ότι υπάρχει μια ακμή μεταξύ των κορυφών  $u$  και  $v$ . Οι κορυφές είναι αριθμημένες

από 1 έως  $n$ . Η Μαρία είναι πάντοτε η κορυφή 1. Η συνάρτησή σας μπορεί να κάνει κλήσεις στις παρακάτω συναρτήσεις:

```
int GetMove();
```

Αυτή η συνάρτηση πρέπει να καλείται κάθε φορά που είναι η σειρά της Μαρίας, όπως στην αρχή του παιχνιδιού. Εάν καλέσετε τη συνάρτηση όταν δεν είναι η σειρά της Μαρίας, θα λάβετε ως αποτέλεσμα `Wrong Answer`. Η συνάρτηση μπορεί να επιστρέψει μια από τις παρακάτω τιμές:

- έναν ακέραιο  $v$ , όπου  $2 \leq v \leq n$ . Αυτό σημαίνει ότι η Μαρία προκαλεί το άτομο με θέση  $v$ . Αυτή είναι πάντα μια επιτρεπτή κίνηση.
- 0, εάν η Μαρία παραιτηθεί από το παιχνίδι. Η Μαρία θα παραιτείται πάντα εάν δεν έχει επιτρεπτές κινήσεις. Όταν συμβαίνει αυτό το πρόγραμμα σας πρέπει να επιτρέψει στην συνάρτηση `SocialEngineering` να επιστρέψει (`return`), και θα λάβετε το αποτέλεσμα `Accepted`.

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

Αυτή η συνάρτηση πρέπει να καλείται κάθε φορά που δεν είναι η σειρά της Μαρίας. Αυτό σημαίνει ότι το άτομο που είναι η σειρά του προκαλεί το άτομο  $v$ . Εάν αυτή δεν είναι μια επιτρεπτή κίνηση, ή είναι η σειρά της Μαρίας την ώρα της κλήσης, τότε θα λάβετε το αποτέλεσμα `Wrong Answer`.

Εάν η Μαρία έχει μια στρατηγική νίκης στην αρχή του παιχνιδιού, το πρόγραμμά σας πρέπει να επιτρέψει στην `SocialEngineering()` να επιστρέψει (`return`) πριν την πρώτη κλήση της `GetMove()`. Τότε θα λάβετε το αποτέλεσμα `Accepted`.

## Περιορισμοί

- $2 \leq n \leq 2 \cdot 10^5$
- $1 \leq m \leq 4 \cdot 10^5$
- Ο γράφος είναι συνδεδεμένος. Κάθε μη ταξινομημένο ζεύγος κορυφών θα εμφανιστεί το πολύ μία φορά ως ακμή, και κάθε ακμή θα υπάρχει μεταξύ δύο διαφορετικών κορυφών.

## Υποπροβλήματα

Η Μαρία θα παίζει πάντα τέλεια με την έννοια ότι θα κάνει νικηφόρες κινήσεις όποτε έχει στρατηγική νίκης. Αν δεν έχει στρατηγική νίκης, τότε θα προσπαθήσει να παρασύρει το πρόγραμμά σας να κάνει λάθος, με διάφορους έξυπνους τρόπους. Θα παραιτηθεί μόνο εάν δεν έχει επιτρεπτές κινήσεις, εκτός από το Υποπρόβλημα 3.

1. (15 βαθμοί)  $n, m \leq 10$ .

2. (15 βαθμοί) Όλοι, εκτός της Μαρίας, έχουν το πολύ 2 φίλους.
3. (20 βαθμοί) Η Μαρία θα παραιτηθεί αμέσως, εκτός αν έχει μια στρατηγική νίκης.
4. (25 βαθμοί)  $n, m \leq 100$ .
5. (25 βαθμοί) Χωρίς επιπλέον περιορισμούς.

## Παράδειγμα Διάδρασης (Sample Interaction)

Η δική σας ενέργεια	Ενέργεια βαθμολογητή	Επεξήγηση
-	<code>SocialEngineering(5, 6, {{1,4}, {1,5}, {2,4}, {2,5}, {2,3}, {3,5}})</code>	<code>SocialEngineering()</code> καλείται με έναν γράφο με 5 κορυφές και 6 ακμές.
<code>GetMove()</code>	Επιστρέφει 4	Η Μαρία προκαλεί το άτομο 4.
<code>MakeMove(2)</code>	-	Το άτομο 4 προκαλεί το άτομο 2.
<code>MakeMove(5)</code>	-	Το άτομο 2 προκαλεί το άτομο 5.
<code>MakeMove(1)</code>	-	Το άτομο 2 προκαλεί την Μαρία.
<code>GetMove()</code>	Επιστρέφει 0	Η Μαρία δεν έχει άλλες επιτρεπτές κινήσεις κι έτσι παραιτείται.
Επιστρέφει (return)	-	Κερδίσατε το παιχνίδι και επιτρέπει στην <code>SocialEngineering()</code> να επιστρέψει (return).

Η δική σας ενέργεια	Ενέργεια βαθμολογητή	Επεξήγηση
-	<code>SocialEngineering(2, 1, {{1,2}})</code>	<code>SocialEngineering</code> καλείται με έναν γράφο με 2 κορυφές και 1 ακμές.
Επιστρέφει (return)	-	Η Μαρία έχει μια στρατηγική νίκης στον γράφο της, κι έτσι πρέπει να επιστρέψετε (return) χωρίς να καλέσετε την <code>GetMove()</code> για να παραιτηθείτε.

## Παράδειγμα Βαθμολογητή

Το παρεχόμενο παράδειγμα βαθμολογητή, `grader.cpp`, στο επισυναπτόμενο αρχείο του προβλήματος `SocialEngineering.zip`, διαβάζει την είσοδο από την τυπική είσοδο (standard input) στην παρακάτω μορφή:

- Η πρώτη γραμμή περιέχει το πλήθος των κόμβων,  $n$ , και το πλήθος των ακμών,  $m$  στον γράφο.
- Οι παρακάτω  $m$  γραμμές περιέχουν ακμές ως ζεύγη κορυφών.

Το παράδειγμα του βαθμολογητή διαβάζει την είσοδο και καλεί την συνάρτηση `SocialEngineering()` στη λύση σας. Σημειώστε ότι το παράδειγμα του βαθμολογητή δεν υλοποιεί την στρατηγική νίκης της Μαρίας και παρέχεται μόνον για παράδειγμα διάδρασης (sample interaction).

Για να κάνετε `compile` το παράδειγμα του βαθμολογητή με την λύση σας, μπορείτε να χρησιμοποιήσετε την παρακάτω εντολή στη γραμμή εντολών (terminal prompt):

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

όπου `solution.cpp` είναι το αρχείο της λύσης σας που θα υποβληθεί στο CMS. Για να τρέξετε το πρόγραμμα με το παράδειγμα εισόδου που παρέχεται στο επισυναπτόμενο, πληκτρολογήστε την παρακάτω εντολή στη γραμμή εντολών (terminal prompt):

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