



Peli

Jian-Jia on pieni poika, joka rakastaa pelien pelaamista. Kun häneltä kysytään kysymys, hän mieluummin tekee siitä pelin kuin vastaa suoraan. Jian-Jia tapasi ystävänsä Mei-Yun ja kertoi hänelle Taiwanin lentoyhteysistä. Taiwanissa on n kaupunkia (numerointina $0, \dots, n - 1$), joista joidenkin välillä on lentoyhteys. Jokainen lentoyhteys yhdistää kaksi kaupunkia ja sitä voi lentää molempiin suuntiin.

Mei-Yu kysyi Jian-Jialta, onko mahdollista matkustaa minkä tahansa kahden kaupungin välillä lentokoneella (joko suoraan tai välikaupunkien kautta). Jian-Jia ei halunnut paljastaa vastausta, mutta sen sijaan ehdotti peliä. Mei-Yu voi kysyä häneltä kysymyksiä muotoa "Onko kaupunkien x ja y välillä *suora* lentoyhteys?" ja Jian-Jia vastaa tällaisiin kysymyksiin välittömästi. Mei-Yu kysyy jokaisesta kaupunkiparista kerran, mistä tulee yhteensä $r = n(n - 1)/2$ kysymystä. Mei-Yu voittaa pelin, jos hän voi päätellä ensimmäisen i kysymyksen vastauksen jälkeen, onko verkosto yhtenäinen vai ei, missä $i < r$. Verkosto on yhtenäinen, jos on mahdollista matkustaa minkä tahansa kahden kaupunkiparin välillä lentämällä (joko suoraan tai välikaupunkien kautta). Jos kuitenkin Mei-Yu tarvitsee kaikki r kysymystä, voittaja on Jian-Jia.

Jotta peli olisi hauskempi Jian-Jialle, ystävykset sopivat, että Jian-Jian ei tarvitse välittää todellisista Taiwanin lentoyhteysistä, vaan hän voi muodostaa verkkoa pelin edetessä Mei-Yun aiempien vastausten perusteella. Tehtäväsi on auttaa Jian-Jiaa voittamaan peli päättämällä, miten hänen tulee vastata kysymyksiin.

Esimerkit

Selitämme pelin säännöt kolmen esimerkin avulla. Jokaisessa esimerkissä on $n = 4$ kaupunkia ja $r = 6$ kierrosta kysymyksiä ja vastauksia.

Ensimmäisessä esimerkissä (seuraava taulukko) Jian-Jia *häviää*, koska kierroksen 4 jälkeen Mei-Yu tietää varmasti, että minkä tahansa kaupungin välillä voi matkustaa lentäen. Tätä ei voi enää muuttaa se, miten Jian-Jia vastaa kysymyksiin 5 ja 6.

kierros	kysymys	vastaus
1	0, 1	kyllä
2	3, 0	kyllä
3	1, 2	ei
4	0, 2	kyllä
-----	-----	-----
5	3, 1	ei
6	2, 3	ei

Seuraavassa esimerkissä Mei-Yu voi osoittaa kierroksen 3 jälkeen, että verkossa *ei ole* mahdollista

matkustaa kaupunkien 0 ja 1 välillä riippumatta Jian-Jian vastauksista kysymyksiin 4, 5 ja 6. Niinpä Jian-Jia häviää jälleen.

kierros	kysymys	vastaus
1	0, 3	ei
2	2, 0	ei
3	0, 1	ei
-----	-----	-----
4	1, 2	kyllä
5	1, 3	kyllä
6	2, 3	kyllä

Viimeisessä esimerkissä Mei-Yu ei pysty tietämään, voiko minkä tahansa kahden kaupungin välillä matkustaa lentäen, ennen kuin kaikkiin kuuteen kysymykseen on tullut vastaus. Tässä tapauksessa Jian-Jia *voittaa* pelin. Tarkemmin ottaen koska Jian-Jia vastasi *kyllä* viimeiseen kysymykseen (seuraavassa taulukossa), on mahdollista matkustaa minkä tahansa kahden kaupungin välillä. Mutta jos Jian-Jia olisi vastannut *ei* viimeiseen kysymykseen, tämä ei olisi mahdollista.

kierros	kysymys	vastaus
1	0, 3	ei
2	1, 0	kyllä
3	0, 2	ei
4	3, 1	kyllä
5	1, 2	ei
6	2, 3	kyllä

Tehtävä

Kirjoita ohjelma, joka auttaa Jian-Jiaa voittamaan pelin. Huomaa, että kumpikaan pelaajista ei tiedä toisen strategiaa. Mei-Yu voi kysyä kaupunkipareja missä tahansa järjestyksessä ja Jian-Jian täytyy vastata heti tietämättä tulevia kysymyksiä. Sinun täytyy toteuttaa seuraavat funktiot:

- `initialize(n)` -- Testaaja kutsuu ensin funktiota `initialize`. Parametri n on kaupunkien määrä.
- `hasEdge(u, v)` -- Tämän jälkeen testaaja kutsuu funktiota `hasEdge` $r = n(n - 1)/2$ kertaa. Nämä kutsut kuvaavat Mei-Yun kysymyksiä siinä järjestyksessä kuin hän kysyy niitä. Sinun täytyy vastata, onko kaupunkien u ja v välillä suoraa lentoa. Tarkemmin palautusarvon tulee olla 1, jos suora yhteys on olemassa, ja muuten 0.

Osatehtävät

Jokainen osatehtävä muodostuu joukosta pelejä. Saat pisteitä osatehtävästä vain silloin, jos ohjelmasi avulla Jian-Jia voittaa jokaisen pelin.

osatehtävä	pisteet	n
1	15	$n = 4$
2	27	$4 \leq n \leq 80$
3	58	$4 \leq n \leq 1500$

Toteutus

Sinun tulee lähettää tarkalleen yksi tiedosto nimeltä `game.c`, `game.cpp` tai `game.pas`. Tämä tiedosto toteuttaa yllä olevat aliohjelmat käyttäen seuraavia runkoja.

C/C++-ohjelmat

```
void initialize(int n);
int hasEdge(int u, int v);
```

Pascal-ohjelmat

```
procedure initialize(n: longint);
function hasEdge(u, v: longint): longint;
```

Esimerkkitarkastin

Esimerkkitarkastin lukee syötteen seuraavassa muodossa:

- rivi 1: n
- seuraavat r riviä: jokaisella rivillä on kaksi kokonaislukua u ja v , jotka tarkoittavat kysymystä liittyen kaupunkeihin u ja v .