



## Igra

Jian-Jia je mladić koji voli igrati igre. Kada mu neko postavi pitanje, on više voli igrati igre nego odgovoriti direktno na postavljeno pitanje. Jian-Jia je sreo svoju prijateljicu Mei-Yu i ispričao joj o mreži letova na Taiwanu. Na Taiwan-u ima ukupno  $n$  gradova (numerisanih sa  $0, \dots, n - 1$ ), od kojih su samo neki direktno povezani letovima. Svaki let povezuje dva grada i može ići u oba pravca.

Mei-Yu je pitala Jian-Jia da li je moguće putovati avionom između svaka dva grada (bilo direktno bilo indirektno). Jian-Jia nije želio direktno dati odgovor, i umjesto toga predložio je da igraju jednu igru.

Mei-Yu mu može postavljati pitanja u sljedećoj formi "Da li su gradovi  $x$  i  $y$  direktno povezani letom?", i Jian-Jia će odmah odgovoriti na takvo pitanje. Mei-Yu će pitati za svaka dva para gradova tačno jednom, što je ukupno  $r = n(n - 1)/2$  mogućih pitanja. Mei-Yu pobjeđuje u igri ako, nakon što dobije odgovore na prvih  $i$  pitanja za neki  $i < r$ , može zaključiti da li je, da ili ne, mreža povezana. Drugim rječima, Mei-Yu će znati da li je moguće putovati između svakog para gradova (direktno ili indirektno). U svim ostalim slučajevima, to jest, ako joj trebaju odgovori na svih  $r$  pitanja, onda je pobjednik Jian-Jia.

Kako bi igra bila još zabavnija za Jian-Jia, prijatelji su se dogovorili da on može zanemariti stvarnu mrežu letova na Taiwan-u. Jian-Jia može izmišljati novu mrežu kako igra napreduje i u zavisnosti od postavljenih pitanja. Vaš zadatak je da pomognete Jian-Jia da pobijedi u igri, odlučujući na koji način treba odgovarati na pitanja.

## Primjeri

Objasnićemo pravila igre na tri primjera. Svaki primjer ima  $n = 4$  gradova i  $r = 6$  krugova pitanja i odgovora.

U prvom primjeru (sljedeća tabela), Jian-Jia *gubi* jer nakon četvrtog kruga, Mei-Yu tačno zna da je moguće letiti između svaka dva grada, bez obzira kako Jian-Jia odgovori na pitanja 5 ili 6.

krug	pitanje	odgovor
1	0, 1	da
2	3, 0	da
3	1, 2	ne
4	0, 2	da
-----	-----	-----
5	3, 1	ne
6	2, 3	ne

U sljedećem primjeru Mei-Yu može dokazati poslije trećeg kruga, bez obzira kako Jian-Jia odgovori na pitanja 4, 5, ili 6, da *nije moguće* putovati između gradova 0 i 1 avionom, tako da Jian-Jia ponovo *gubi*.

krug	pitanje	odgovor
1	0, 3	ne
2	2, 0	ne
3	0, 1	ne
----	-----	-----
4	1, 2	da
5	1, 3	da
6	2, 3	da

U posljednjem primjeru Mei-Yu ne može utvrditi da li se može letiti između bilo koja dva grada nakon odgovora na svih šest pitanja, tako da Jian-Jia *pobjeđuje* ovu igru. Naime, pošto je Jian-Jia odgovorio *da* na posljednje pitanje (u tabeli), onda je moguće putovati između bilo koja dva grada. Međutim, da je Jian-Jia odgovorio *ne* na posljednje pitanje, onda bi to bilo nemoguće.

krug	pitanje	odgovor
1	0, 3	ne
2	1, 0	da
3	0, 2	ne
4	3, 1	da
5	1, 2	ne
6	2, 3	da

## Zadatak

Molim Vas, napišite program koji pomaže Jian-Jia da pobijedi u navedenoj igri. Imajte na umu da ni Mei-Yu ni Jian-Jia ne zna strategiju onog drugog. Mei-Yu može pitati za parove gradova po bilo kom redosljedu, i Jian-Jia mora odmah odgovoriti bez poznavanja budućih pitanja. Morate realizovati slijedeće dvije funkcije.

- `initialize(n)` -- Prvo će biti pozvana funkcija `initialize`. Parametar  $n$  je broj gradova.
- `hasEdge(u, v)` -- Zatim ćemo pozvati `hasEdge`  $r = n(n - 1)/2$  puta. Ovi pozivi predstavljaju Mei-Yu pitanja, u onom redosljedu u kojem su postavljena. Morate odgovoriti da li postoji direktan let između gradova  $u$  i  $v$ . Naime, odgovor treba biti 1 ako postoji direktan let, ili 0 ako ne postoji direktan let.

## Podzadaci

Svaki podzadatak se sastoji od nekoliko igara. Vi ćete dobiti poene samo za podzadatak u kome vaš program pobjeđuje sve igre za Jian-Jia.

podzadatak	poeni	$n$
1	15	$n = 4$

podzadatak	poeni	$n$
2	27	$4 \leq n \leq 80$
3	58	$4 \leq n \leq 1500$

## Detalji implementacije

Trebate submitovati tačno jedan file pod nazivom `game.c`, `game.cpp` or `game.pas`. Ovaj file implementira podprogram kako je gore opisano koristeći sljedeći tip funkcije.

### C/C++ programs

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

### Pascal programs

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

### Sample grader

Grader čita ulazne podatke date u sljedećem formatu:

- line 1:  $n$
- the following  $r$  lines: each line contains two integers  $u$  and  $v$  that describe a question regarding cities  $u$  and  $v$ .