



Igrica

Jian-Jia (Janez) je dečko, ki obožuje igrice vseh vrst. Ko dobi vprašanje, namesto da bi odgovoril neposredno, si raje izmisli igrico. Jian-Jia je srečal svojo prijateljico Mei-Yu (Metka) in ji razložil vse o omrežju poletov med mesti na Tajvanu. Na Tajvanu je n mest, ki so oštevilčena od 0 do $n - 1$. Nekatera izmed teh mest so povezna z letalskimi linijami. Vsaka linija povezuje dve mesti in letimo lahko v obe smeri.

Mei-Yu je vprašala Jian-Jia ali je mogoče potovati z letalom med katerima koli mestoma (lahko neposredno ali posredno). Nesramen kot je, ji Jian-Jia seveda ni hotel razkriti odgovora, ampak je predlagal, da Mei-Yu lahko postavlja vprašanja oblike: "Ali sta mesti x in y povezani z *neposredno* letalsko linijo?". Na taka vprašanja bo Jian-Jia takoj odgovoril. Mei-Yu bo vprašala tako vprašanje za vse pare mest, vse skupaj torej $r = n(n - 1)/2$ vprašanj. Mei-Yu v tej igri zmagaja, če lahko po prvih i odgovorih na njena vprašanja (pri $i < r$) ugotovi, ali je mogoče leteti med katerimakoli mestoma (posredno ali neposredno) ali če ugotovi, da se tega ne da. Če pa potrebuje kar vseh r vprašanj, potem je zmagovalec Jian-Jia.

Da bi bila igra Jian-Jia bolj zabavna, se prijatelja dogovorita, da bosta ignorirala letalsko omrežje Tajvana in se Jian-Jia lahko kar zmišljuje omrežje tekom same igre in izbira odgovore glede na predhodna vprašanja. Tvoja naloga je, da pomagaš Jian-Jia v tej igri zmagati. Ugotoviti moraš, kako naj Jian-Jia odgovarja, da bo Mei-Yu potrebovala vseh r vprašanj, da ugotovi ali je omrežje povezano.

Primeri

Igrico bomo razložili s tremi primeri. Vsak primer ima $n = 4$ mesta in $r = 6$ izmenjav vprašanj in odgovorov.

V prvem primeru (spodnja tabela) naš Jian-Jia *izgubi* igro, ker po četrti izmenjavi Mei-Yu natanko ve, da je mogoče potovati med vsemi mesti, neodvisno od tega kako Jian-Jia odgovori na vprašanja 5 in 6.

izmenjava	vprašanje	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

V naslednjem primeru lahko Mei-Yu po izmenjavi 3 dokaže, da se ne da potovati med mestoma 0 in 1, ne glede na to kako Jian-Jia odgovori na naslednja tri vprašanja. Jian-Jia je torej tudi v tem primeru izgubil.

izmenjava	vprašanje	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

V zadnjem primeru pa si pogledjmo zmago Jian-Jia. Mei-Yu namreč ne more povedati ali je omrežje povezano, vse dokler ne vpraša čisto vseh šestih vprašanj.

Ker je Jian-Jia na zadnje vprašanje odgovoril *da*, vemo, da je letalsko potovanje med vsemi mesti mogoč, če pa bi bil njegov odgovor *ne*, bi bilo to omrežje nepovezano.

izmenjava	vprašanje	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

Naloga

Napiši program, ki bo pomagal zmagati Jian-Jia. Upoštevaj, da niti eden, niti drugi ne poznata strategije svojega nasprotnika. Mei-Yu lahko povpraša o katerem koli paru mest v katerem koli vrstnem redu. In Jian-Jia mora na ta vprašanja odgovoriti tako, ne da bi vedel sledeča vprašanja. Implementirati moraš sledeči dve funkciji:

- `initialize(n)` -- Funkcijo `initialize` bomo poklicali najprej. Parameter n je število mest.
- `hasEdge(u, v)` -- Za inicijalizacijo bomo poklicali `hasEdge` $r = n(n - 1)/2$ -krat. Ti klici predstavljajo vprašanja, ki jih postavlja Mei-Yu. Ta funkcija odgovarja ali obstaja neposredna letalska linija med mesti u in v . Vrnjena vrednost mora biti 1, če neposredna linija obstaja in 0, če linija ne obstaja.

Podnaloge

Vsaka podnaloga se sestoji iz več iger. Točke za eno podnalogo boste dobili samo, če vaš program zmaga v vseh igrah.

podnaloga	točke	n
1	15	$n = 4$
2	27	$4 \leq n \leq 80$
3	58	$4 \leq n \leq 1500$

Podrobnosti implementacije

Oddati moraš natanko eno datoteko poimenovano `game.c`, `game.cpp` ali `game.pas`. V tej datoteki morajo biti implementirane funkcije natanko tako, kot so opisane zgoraj.

Podpisi zahtevanih funkcij so sledeči:

V programskem jeziku C/C++

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

V programskem jeziku Pascal

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

Vzorčni ocenjevalnik

Vzorčni ocenjevalnik bere vhod v sledečem formatu:

- vrstica 1: n
- sledečih r vrstic: vsaka vrstica vsebuje dve celi števili u in v , ki opisujeta vprašanje o mestih u in v .