



Játék

Jian-Jia és Mei-Yu Taiwan repülő hálózata alapján játszik. Van n város (0-tól $n - 1$ -ig sorszámozva), néhányat kétirányú repülő járat köt össze.

A játékban Mei-Yu kérdez, a kérdése formája: "Van-e az x . és y . város között *közvetlen* repülő járat?". A kérdésekre Jian-Jia azonnal válaszol. Mei-Yu minden várospárra pontosan egyszer kérdez rá, így összesen $r = n(n - 1)/2$ kérdést tesz fel. Mei-Yu nyer, ha r -nél kevesebb kérdés feltétele után el tudja dönteni, hogy bármely városból el lehet-e jutni bármely másik városba közvetlen vagy közvetett úton. Egyébként, azaz ha csak r kérdés feltétele után tudja eldönteni, akkor Jian-Jia nyer.

Segíts Jian-Jia-nak megnyerni a játékot!

Példa

Mindhárom példában $n = 4$ város és $r = 6$ kérdés van, amire válaszolnod kell.

Az első példában (az alábbi táblában) Jian-Jia *veszít*, mert a 4. kérdés után Mei-Yu tudja, hogy bármely városból bármely városba el lehet jutni, függetlenül az 5. és a 6. kérdésre adott választól.

lépés	kérdés	válasz
1	0, 1	igen
2	3, 0	igen
3	1, 2	nem
4	0, 2	igen
-----	-----	-----
5	3, 1	nem
6	2, 3	nem

A 2. példában Mei-Yu 3 kérdés után tudja, hogy nem lehet bárhonnán bárhova eljutni, például a 0. városból az 1. városba sem. Tehát Jian-Jia *veszít*.

lépés	kérdés	válasz
1	0, 3	nem
2	2, 0	nem
3	0, 1	nem
-----	-----	-----
4	1, 2	igen

lépés	kérdés	válasz
5	1, 3	igen
6	2, 3	igen

Az utolsó példában Mei-Yu csak 6 kérdés után tudja a választ, tehát Jian-Jia *nyeri* a játékot. Ez azért van, mert ha Jian-Jia válasza az utolsó kérdésre *igen*, akkor el lehet jutni bárhonnán bárhova, ha pedig a válasza *nem*, akkor ez lehetetlen.

lépés	kérdés	válasz
1	0, 3	nem
2	1, 0	igen
3	0, 2	nem
4	3, 1	igen
5	1, 2	nem
6	2, 3	igen

Feladat

Írj programot, amellyel Jian-Jia nyer! Tudjuk, hogy Mei-Yu és Jian-Jia nem ismerik egymás stratégiáját. Mei-Yu tetszőleges sorrendben kérdezhet, Jian-Jia-nak pedig azonnal válaszolnia kell! A következő két függvényt kell megvalósítanod:

- `initialize(n)` -- Az `initialize`-t az értékelő egyszer hívja, a legelején. Az n paraméterben adja meg a városok számát.
- `hasEdge(u, v)` -- Ezután a `hasEdge` függvényt $r = n(n - 1)/2$ -szer hívja meg. A hívások Mei-Yu kérdéseit valósítják meg, amire meg kell adnod, hogy van-e u és v között közvetlen repülő járat. A függvény értéke 1 legyen, ha van közvetlen járat, illetve 0, ha nincs!

Részfeladatok

részfeladat	pont	n
1	15	$n = 4$
2	27	$4 \leq n \leq 80$
3	58	$4 \leq n \leq 1500$

Megvalósítás

A `game.c`, `game.cpp` vagy `game.pas` fájlt kell beküldened! Ebben legyen a két megvalósított függvény! A `game.h`-t include-old!

C/C++ program

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

Pascal program

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

Minta értékelő

A minta értékelő a bemenetet a következő formában várja:

- 1. sor: n
- A következő r sor mindegyike egy u és v számpárt tartalmaz, a kérdésben szereplő városok u és v sorszámát.