



Game

Когато задават въпроси на Джан, той предпочита да играе игра, вместо да отговори директно. Джан среща приятелката си Мян и започва да ѝ говори за мрежата от самолетни връзки в Тайван. Има n града (номерирани от 0 до $n - 1$), някои от които са свързани с директни самолетни полети. Всеки директен полет свързва два града и може да бъде взет във всяка от двете посоки.

Мян пита Джан дали е възможно да отиде от всеки град до всеки друг със самолет (или директно или чрез прекачаване). Джан не иска да ѝ даде отговор, а предлага да играят игра: Мян да му задава въпроси от вида „Свързани ли са градовете x и y с директен полет?“ и Джан ще отговоря веднага на въпроса. Мян ще задава въпрос за всяка двойка градове, т.е. броят на въпросите ѝ ще бъде $r = n(n - 1)/2$. Мян ще победи в играта, ако след получаване на отговорите за първите i въпроса (за някое $i < r$) тя ще може да определи дали е възможно или не е възможно да се лети от всеки град до всеки друг (или директно или чрез прекачаване). Мян губи играта (т.е. Джан печели), ако трябва да зададе всичките r въпроса.

За да бъде играта по-забавна за Джан, Мян се съгласява Джан да не използва реалните самолетни полети в Тайван, а да измисля мрежата от полети по време на играта, като избира своите отговори в зависимост от предишните въпроси на Мян. Вашата задача е да помогнете на Джан да победи в играта, като му казвате как да отговаря на въпросите.

Примери

Ще обясним играта с 3 примера. Всеки пример има $n = 4$ града и $r = 6$ хода.

Пример 1. Джан губи играта след ход 4, защото Мян вече знае, че е възможно да се лети от всеки град до всеки друг и е без значение как Джан ще отговори на въпроси 5 и 6.

ход	въпрос	отговор
1	0, 1	yes
2	3, 0	yes
3	1, 2	no
4	0, 2	yes
-----	-----	-----
5	3, 1	no
6	2, 3	no

Пример 2. Мян може да е сигурна, че след ход 3, независимо от отговорите на Джан за въпроси 4, 5 и 6, тя не може да лети между градове 0 и 1, така че Джан губи играта.

ход	въпрос	отговор
-----	--------	---------

ход	въпрос	отговор
1	0, 3	no
2	2, 0	no
3	0, 1	no
----	-----	-----
4	1, 2	yes
5	1, 3	yes
6	2, 3	yes

Пример 3. Мян не може да определи дали е възможно или не да се лети от всеки град до всеки друг, преди да е задала всичките 6 въпроса. Така Джан е победител. Освен това, понеже Джан отговаря yes на последния въпрос, възможно е да се лети от всеки град до всеки друг. Ако Джан беше отговорил no на този въпрос, нямаше да е възможно да се лети от всеки град до всеки друг.

ход	въпрос	отговор
1	0, 3	no
2	1, 0	yes
3	0, 2	no
4	3, 1	yes
5	1, 2	no
6	2, 3	yes

Задача

Напишете програма, която помага на Джан да победи в играта. Нито Мян, нито Джан знаят стратегията на другия. Мян може да запитва за двойки градове в произволен ред и Джан трябва да отговаря веднага, без да знае какви ще са следващите въпроси. Трябва да програмирате следните две функции:

- `initialize(n)` -- Ние извикваме първо тази функция. Стойността n е броят на градовете.
- `hasEdge(u, v)` -- Ние извикваме тази функция $r = n(n - 1)/2$ пъти. Всяко извикване представя въпрос на Мян, в реда, в който го задава. Вие трябва да отговорите дали има директен полет между градове u и v (отговаряте 1, ако има такъв или 0, в обратния случай).

Подзадачи

подзадача	точки	n
1	15	$n = 4$
2	27	$4 \leq n \leq 80$
3	58	$4 \leq n \leq 1500$

Описание на имплементацията на C/C++:

Трябва да изпратите само един файл с име `game.c` или `game.cpp`. Този файл трябва да съдържа имплементацията на функциите, описани по-горе те трябва да имат следната сигнатура:

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

Примерен грейдър

Той чете вход със следния формат:

- На първия ред е записано числото n .
- Всеки от следващите r реда съдържа по две цели числа u и v , които определят въпрос, отнасящ се за градовете u и v .