



## 猜謎遊戲

建嘉是個喜歡玩猜謎的青年。當有人問他問題時，他不喜歡直接回答，而喜歡跟對方玩猜謎。有天建嘉遇到他的朋友美玉，並跟她談到臺灣的飛航網路。臺灣有  $n$  個城市（編號為  $0, \dots, n-1$ ），而有些城市間存在飛機航線。每個航線連接兩個城市，而且是雙向的。

美玉問建嘉，是不是所有城市都能直接或間接地搭飛機往來。建嘉不想直接說出答案，所以提議玩個猜謎遊戲。美玉可以問他「城市  $x$  和  $y$  是否有『直接』的航線？」，而建嘉會立刻回答是或不是。美玉會試圖詢問每一組城市是否存在直接航線，每組詢問一次，總共問  $r = n(n-1)/2$  個問題。如果美玉問到第  $i$  ( $i < r$ ) 個問題就能推論出網路是否為連通（即「是否所有城市都能直接或間接搭機往來」），美玉就贏了。反過來，如果美玉需要問完所有  $r$  個問題才能知道答案，那麼建嘉就贏了。

爲了讓遊戲更有趣點，這對朋友同意建嘉不需在意真實的臺灣飛航網路，而建嘉可以在猜謎遊戲過程中建構他的網路，依據美玉之前的提問來決定後面的回答。你的任務是幫助建嘉決定該怎麼回答，才能贏得遊戲。

## 範例

我們以下面三個例子說明遊戲規則。每個例子有  $n = 4$  個城市、 $r = 6$  個問題和回答。

在第一個例子（下方表格）中，建嘉輸了。因爲美玉問完前 4 個問題，不管第 5 或 6 個問題的回答，她已經能確知「所有城市都能直接或間接搭機往來」。

round	question	answer
1	0, 1	yes
2	3, 0	yes
3	1, 2	no
4	0, 2	yes
-----	-----	-----
5	3, 1	no
6	2, 3	no

在下一個例子中，美玉問完前 3 個問題，不管第 4, 5, 6 個問題的回答，她已經能確知城市 0 和城市 1 「不可能」透過飛機往來。因此，建嘉又輸了。

round	question	answer
1	0, 3	no
2	2, 0	no
3	0, 1	no
-----	-----	-----
4	1, 2	yes
5	1, 3	yes
6	2, 3	yes

在最後一個例子中，美玉在問完所有 6 個問題前，都無法確知「是否所有城市都能直接或間接搭機往來」，所以這次建嘉贏了。更明確地來說，如果最後一個問題建嘉回答「是」（下表的狀態），那麼所有城市都能直接或間接搭機往來；然而，如果建嘉回答「否」，則不是所有城市都能直接或間接往來。

round	question	answer
1	0, 3	no
2	1, 0	yes
3	0, 2	no
4	3, 1	yes
5	1, 2	no
6	2, 3	yes

## 任務

請寫個程式幫助建嘉贏得遊戲。要注意美玉和建嘉並無法知道彼此的策略。美玉可以用任何順序來詢問城市間是否存在直接航線，而建嘉必須立即回答。他無法知道美玉接下來會如何詢問問題。你必須實作以下兩個函式：

- `initialize(n)` -- 評分程式會先呼叫你的 `initialize` 函式。參數  $n$  代表城市個數。
- `hasEdge(u, v)` -- 接著會呼叫 `hasEdge`  $r$  次， $r = n(n - 1)/2$ 。這些呼叫代表美玉依序詢問的問題。這個函式必須回答城市  $u$  和  $v$  之間是否存在直接航線。更明確地說，如果存在直接航線，回傳 1；否則，回傳 0。

## 子任務

每個子任務會有好幾場遊戲。只有當所有遊戲都是建嘉獲勝時，你才可獲得該子任務的分數。

subtask	points	$n$
1	15	$n = 4$
2	27	$4 \leq n \leq 80$

subtask	points	$n$
3	58	$4 \leq n \leq 1500$

## 實作細節

你必須上傳單一個檔案，檔名為 `game.c`、`game.cpp` 或者 `game.pas`。這個檔案裡實作了上述的兩個函式，其函式原型如下：

### 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

範例評分程式會讀入檔案，格式如下：

- 第 1 行:  $n$
- 接下來的  $r$  行: 每一行有兩個整數  $u$  和  $v$ ，代表詢問城市  $u$  和  $v$  是否有直接航線。