



## Гра

Жиан-Жиа це маленький хлопчик, який полюбляє грати в ігри. Коли його щось запитують, від воліє почати грати, а не відповісти прямо. Жиан-Жиа зустрів свою подругу Мей-Ю та розповів їй про мережу перельотів у Тайвані. У Тайвані є  $n$  міст (пронумерованих  $0, \dots, n - 1$ ), деякі з них з'єднано перельотами. Кожен переліг з'єднує два міста та може виконуватись у двох напрямках.

Мей-Ю запитала Жиан-Жиа, чи можна потрапити з одного міста до іншого літаком (прямо чи ні). Жиан-Жиа не хоче прозкривати відповідь, замість цього він пропонує зіграти у гру. Мей-Ю може задавати запитання вигляду "Чи з'єднані міста  $x$  та  $y$  *прямим* перельотом?", а Жиан-Жиа буде негайно на такі питання відповідати. Мей-Ю буде запитувати про кожну пару міст тільки один раз, задавши всього  $r = n(n - 1)/2$  запитань. Мей-Ю виграє гру, якщо, отримавши відповіді на перші  $i$  запитань для деякого  $i < r$ , вона зможе зробити висновок, чи є мережа зв'язною, тобто чи можна подорожувати літаком між довільною парою міст (прямо чи через інші міста). У іншому випадку, тобто якщо їй потрібно всі  $r$  запитань, переможцем буде Жиан-Жиа.

Щоб зробити гру більш цікавою для Жиан-Жиа, друзі домовились забути про справжню мережу перельотів у Тайвані, та вгадувати мережу в процесі гри, обираючи відповідь, базуючись на попередніх запитаннях Мей-Ю. Ваша задача - допомогти Жиан-Жиа виграти гру, вирішуючи, як він має відповідати на запитання.

## Приклади

Пояснимо правила гри на трьох прикладах. Кожен приклад має  $n = 4$  міст та  $r = 6$  раундів запитання-відповідь.

У першому прикладі (наступна таблиця), Жиан-Жиа *програє* оскільки після раунда 4 Мей-Ю знає напевно, що можна подорожувати літаком між довільною парою міст, незалежно від того, як Жиан-Жиа буде відповідати на запитання 5 та 6.

раунд	питання	відповідь
1	0, 1	так
2	3, 0	так
3	1, 2	ні
4	0, 2	тік
-----	-----	-----
5	3, 1	ні
6	2, 3	ні

У наступному прикладі Мей-Ю може довести після раунда 3, що, незалежно від відповідей

Жиан-Жиа на запитання 4, 5 або 6, *немає* можливості подорожувати літаком між містами 0 та 1, Отже Жиан-Жиа знову програє.

раунд	питання	відповідь
1	0, 3	ні
2	2, 0	ні
3	0, 1	ні
-----	-----	-----
4	1, 2	так
5	1, 3	так
6	2, 3	так

У наступному прикладі Мей-Ю не може визначити чи можна подорожувати літаками між довільною парою міст поки не буде отримано відповіді на всі шість запитань, отже Жиан-Жиа *виграє* цю гру. А саме, оскільки Жиан-Жиа відповів *так* на останнє запитання (у наступній таблиці), то можна подорожувати літаком між довільною парою міст. Однак, якби Жиан-Жиа відповів на останнє запитання *ні*, це було б неможливо.

раунд	питання	відповідь
1	0, 3	ні
2	1, 0	так
3	0, 2	ні
4	3, 1	так
5	1, 2	ні
6	2, 3	так

## Задача

Напишіть програму, що допоможе Жиан-Жиа перемогти у грі. Зауважте, що ні Мей-Ю, ні Жиан-Жиа не знають стратегію один одного. Мей-Ю може запитувати про пари міст у довільному порядку, та Жиан-Жиа має негайно відповідати, не знаючи наступних запитань. Ви маєте реалізувати наступні дві функції.

- `initialize(n)` -- Ми спочатку визвемо вашу функцію `initialize`. Параметр  $n$  - це кількість міст.
- `hasEdge(u, v)` -- Потім ми визвемо `hasEdge`  $r = n(n - 1)/2$  раз. Ці виклики подають запитання Мей-Ю, у порядку як вона їх задає. Ви маєте відповідати, чи існує прямий переліт між містами  $u$  та  $v$ . Значення, що повертається, має бути 1, якщо прямий переліт існує, та 0 у іншому випадку.

## Підзадачі

Кожна підзадача складається з кількох ігор. Ви отримаєте бали за підзадачу тільки якщо ваша

програма переможе у всіх іграх за Жиан-Жиа.

підзадача	бали	$n$
1	15	$n = 4$
2	27	$4 \leq n \leq 80$
3	58	$4 \leq n \leq 1500$

## Деталі реалізації

Ви маєте відіслати тільки один файл, що має ім'я `game.c`, `game.cpp` or `game.pas`. Цей файл реалізує підпрограми, що описано вище, використовуючи такі сигнатури.

### Програми на C/C++

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

### Програми на Pascal

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

### Приклад модуля перевірки

Наданий вам модуль перевірки читає вхідні дані у наступному форматі:

- рядок 1:  $n$
- наступні  $r$  рядків: кожен рядок містить два цілих числа  $u$  та  $v$ , що описують запитання відносно міст  $u$  та  $v$ .