



Game

Цзянь-Цзя - маленький мальчик, который любит играть в игры. Когда ему задают вопрос, он предпочитает играть в игры вместо того, чтобы ответить прямо. Цзянь-Цзя встретил свою подругу Мэй-Ю и рассказал ей о схеме полетов в Тайване. Есть n городов в Тайване (пронумерованные $0, \dots, n - 1$), некоторые из которых связаны полетами. Каждый рейс соединяет два города и может быть в обоих направлениях.

Мэй-Ю спросил Цзянь-Цзя, возможно ли пролететь между любыми двумя городами на самолете (напрямую или через другие города). Цзянь-Цзя не захотел отвечать, и вместо этого предложил сыграть в игру. Мэй-Ю может задавать ему вопросы вида "Связаны ли города x и y полетами *напрямую*?", и Цзянь-Цзя немедленно будет отвечать на такие вопросы. Мэй-Ю будет спрашивать о каждой паре городов только один раз, задавая в сумме $r = n(n - 1)/2$ вопросов. Мэй-Ю выигрывает игру, если, после получения ответов на первое i вопросов для некоторого $i < r$, она может определить, соединена ли сеть полетов, то есть возможно ли путешествовать между любыми двумя городами (напрямую или через другие города). Иначе, если она использует все r вопросов, тогда победитель - Цзянь-Цзя.

Для того чтобы игра была более интересной для Цзянь-Цзя, друзья условились, что он может забыть о реальной Тайваньской сети полетов и построить сеть в течение по ходу игры, выбирая его ответы, основанные на предыдущих ответах Мэй-Ю. Ваша задача состоит в том, чтобы помочь Цзянь-Цзя выиграть игру, решая, как он должен ответить на вопросы.

Примеры

Объясним правила игры при помощи трех примеров. В каждом примере $n = 4$ городам и $r = 6$ попыток вопроса и ответа.

В первом примере (нижеследующая таблица), Цзянь-Цзя *проигрывает*, потому что после 4 попытки, Мэй-Ю знает точно, что можно путешествовать между любыми двумя городами полетами, независимо от того как Цзянь-Цзя ответит на вопросы 5 или 6.

Попытка	Вопрос	Ответ
1	0, 1	yes
2	3, 0	yes
3	1, 2	no
4	0, 2	yes
-----	-----	-----
5	3, 1	no
6	2, 3	no

В следующем примере Мэй-Юй может доказать после третьей попытки, независимо от того,

как Цзянь-Цзя отвечает на вопросы 4, 5, или 6, что *невозможно* путешествие полетами между городами 0 и 1; таким образом, Цзянь-Цзя проигрывает снова.

Попытка	Вопрос	Ответ
1	0, 3	no
2	2, 0	no
3	0, 1	no
-----	-----	-----
4	1, 2	yes
5	1, 3	yes
6	2, 3	yes

В заключительном примере Мэй-Ю не может определить, можно ли путешествовать между какими-либо двумя городами полетами, пока на все шесть вопросов не будут даны ответы, таким образом, Цзянь-Цзя *выигрывает* игру. Определенно, если Цзянь-Цзя ответил *да* на последний вопрос (в следующей таблице), тогда возможно путешествовать между любой парой городов. Однако, если бы Цзянь-Цзя ответил *нет* на последний вопрос, то тогда это было бы невозможно.

Попытка	Вопрос	Ответ
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

Пожалуйста, напишите программу, которая поможет Цзянь-Цзя выиграть игру. Обратите внимание на то, что ни Мэй-Ю, ни Цзянь-Цзя не знают стратегию друг друга. Мэй-Ю может спрашивать о парах городов в любом порядке, и Цзянь-Цзя должен немедленно ответить на вопросы, не зная будущих вопросов. Вы должны выполнить следующие две функции.

- `initialize(n)` -- Сначала мы будем вызывать вашу функцию `initialize`. Параметр n - число городов.
- `hasEdge(u, v)` -- Затем мы вызовем функцию `hasEdge` $r = n(n - 1)/2$ раз. Эти вызовы функции представляют вопросы Мэй-Ю в порядке, котором она их задает. Вы должны ответить, есть ли прямой рейс между городами u и v . Определенно, возвращаемое значение должно быть 1, если есть прямой рейс, или 0 иначе.

Подзадачи

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

Детали выполнения

Вы можете послать только один файл, названный `game.c`, `game.cpp` или `game.pas`. Этот файл выполняет подпрограммы, используя следующие объявления.

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;
```

Пример для оценивающей программы (grader)

Ввод данных считывается оценивающей программой в следующем формате:

- строка 1: n
- следующие r строк: каждая строка содержит два целых числа u and v которые описывают вопрос относительно городов u and v .