



بازی

جیان-جیا پسر جوانی است که عاشق بازی‌های فکری است. وقتی از او سؤالی پرسیده می‌شود، ترجیح می‌دهد به جای دادن پاسخ مستقیم یک بازی انجام دهد. او دوستش می-یو را ملاقات کرد و در مورد شبکه‌ی پروازی در تایوان به او توضیح داد. در تایوان n شهر که با شماره‌های 0 تا $n-1$ شناخته می‌شوند وجود دارد که بعضی از آن‌ها با پرواز مستقیم به یکدیگر وصل شده‌اند. هر خط پروازی دو شهر را به صورت دو طرفه به یکدیگر وصل می‌کند.

می-یو از جیان-جیا می‌پرسد که آیا امکان رفتن بین هر دو شهر (با پرواز مستقیم یا غیرمستقیم) وجود دارد؟ جیان-جیا که نمی‌خواهد صریح جواب دهد، درخواست می‌کند که یک بازی انجام دهند. می-یو می‌تواند سؤالی از این نوع بپرسد: «آیا شهرهای x و y با یک پرواز مستقیم به یکدیگر وصل هستند؟». جیان-جیا این نوع سؤالی را بلافاصله با «بله» یا «خیر» پاسخ می‌دهد. می-یو می‌تواند در مورد هر زوج شهر دقیقاً یک سؤال و در مجموع $r = n(n-1)/2$ سؤال بپرسد. می-یو برنده‌ی بازی می‌شود اگر به ازای یک مقدار $r < i$ با دریافت پاسخ i سؤال متوجه شود که آن شبکه همبند است یا خیر؛ به این معنی که آیا امکان رفتن از هر شهری به هر شهر دیگر وجود دارد یا خیر. در غیر این صورت (حالتی که می-یو به r سؤال نیاز پیدا کند) برنده‌ی بازی جیان-جیا خواهد بود.

برای این که بازی برای جیان-جیا جذاب‌تر شود، دو دوست توافق کردند که جیان-جیا شبکه‌ی پروازی تایوان را فراموش کند و در حین بازی شبکه را متناسب با سؤالی‌های می-یو (سؤالی‌های که تا کنون پرسیده شده است) ترسیم کند. وظیفه‌ی شما آن است که به جیان-جیا کمک کنید تا به گونه‌ای به سؤالی‌ها پاسخ دهد که برنده‌ی بازی شود.

مثال

قواعد بازی را با سه مثال توضیح می‌دهیم. هر مثال شامل $n = 4$ شهر و $r = 6$ مرحله سؤال و پاسخ است. در اولین مثال (جدول زیر)، جیان-جیا بعد از چهار مرحله می‌بازد، چرا که می-یو بعد از مرحله‌ی ۴ام مطمئن خواهد بود که شبکه همبند است صرف نظر از آنکه جیان-جیا به سؤالی‌های ۵ و ۶ چه پاسخی دهد.

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

در مثال بعدی، می-یو بعد از مرحله‌ی سوم می‌تواند اثبات کند هر طور جیان-جیا به سؤالی‌های ۴، ۵ و ۶ پاسخ دهد نمی‌توان از شهر ۰ به شهر ۱ رفت. بنابراین جیان-جیا بازی را دوباره می‌بازد.

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

در مثال آخر، می-یو تا پاسخ هر ۶ سؤال را دریافت نکند نمی‌تواند مشخص کند که شبکه همبند است یا خیر؛ بنابراین جیان-جیا برنده‌ی بازی خواهد بود. به‌طور مشخص، چون جیان-جیا به سؤال ۶ پاسخ «بله» داده است (طبق جدول زیر) شبکه همبند است. اگر پاسخ به این سؤال را «خیر» داده بود شبکه ناهمبند می‌شد.

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

مسئله

برنامه‌ای بنویسید که به جیان-جیا کمک کند تا برنده‌ی بازی شود. توجه کنید که می-یو و جیان-جیا از استراتژی یک‌دیگر مطلع نیستند. می-یو می‌تواند به هر ترتیب دل‌خواه در مورد زوج شهرها سؤال کند و جیان-جیا باید بلافاصله بعد از هر سؤال (بدون اطلاع از سؤال‌های بعدی می-یو) به او پاسخ «بله» یا «خیر» دهد. شما باید دو تابع زیر را پیاده‌سازی کنید.

- `initialize(n)`: ما ابتدا تابع `initialize` شما را فراخوانی می‌کنیم. پارامتر n تعداد شهرها است.
- `hasEdge(u, v)`: سپس ما تابع `hasEdge` را $r = n(n-1)/2$ بار فراخوانی می‌کنیم. این فراخوانی‌ها پاسخ سؤال‌های می-یو را به ترتیبی که پرسیده شده‌اند، مشخص می‌کند. شما باید مشخص کنید که بین شهرهای u و v پرواز مستقیم وجود دارد یا خیر. به‌طور مشخص، اگر پرواز مستقیم بین این دو شهر وجود داشت این تابع باید عدد ۱ و در غیر این صورت عدد ۰ را برگرداند.

زیرمسئله‌ها

هر زیرمسئله شامل چندین بازی است. شما امتیاز هر زیرمسئله را به شرطی دریافت می‌کنید که همه‌ی بازی‌ها را به نفع جیان-جیا تمام کنید.

subtask	point	n
1	15	$n = 4$
2	27	$4 \leq n \leq 80$
3	58	$4 \leq n \leq 1500$

جزئیات پیاده‌سازی

شما باید دقیقاً یک فایل به نام `game.c`، `game.cpp` یا `game.pas` ارسال کنید. در این فایل زیربرنامه‌های فوق باید به شکل زیر پیاده‌سازی شوند.

برنامه‌ی C/C++

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

برنامه‌ی پاسکال

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

ارزیاب نمونه

ارزیاب نمونه ورودی را در قالب زیر می‌خواند.

• خط ۱: n

• در هر یک از r خط بعد: هر خط شامل دو عدد صحیح u و v است که به معنی سؤال در مورد شهرهای u و v است.