



## Oyun

Jian-Jia oyun oynamaya bayılan bir gençtir: öyle ki, bir soru sorulduğunda soruyu doğrudan yanıtlamak yerine, oyun oynayarak yanıtlamayı tercih eder. Jian-Jia bugün arkadaşı Mei-Yu ile buluştuğunda O'na Tayvan'ın uçuş ağını anlattı. Tayvan'da  $0, \dots, n - 1$  aralığında numaralanmış  $n$  şehir ve bunların bazıları arasında uçuşlar vardır. Her bir uçuş çift yönlü olup iki şehri birbirine bağlar.

Mei-Yu herhangi iki şehir arasında (doğrudan ya da başka şehirler üzerinden aktarmalı) uçakla gidilip gidilemeyeceğini Jian-Jia'ye sordu. Jian-Jia, cevabı doğrudan söylemek istemedi ve cevabı bulmak için bir oyun oynamayı önerdi. Mei-Yu, " $x$  ve  $y$  şehirleri arasında *doğrudan* bir uçuş var mı?" tipinden sorular sorabilecek ve Jian-Jia her bir soruyu anında cevaplayacaktır. Mei-Yu, her bir şehir çifti için tam olarak bir soru soracak ve toplamda  $r = n(n - 1)/2$  soru sormuş olacaktır. Mei-Yu oyunu şu durumda kazanır:  $i < r$  olacak şekilde herhangi bir  $i$  soru-cevap raundu sonunda uçuş ağının bağlı (connected) olup olmadığını, yani herhangi iki şehir arasında (doğrudan ya da aktarmalı) bir uçuş olup olmadığını kesin olarak bilebilirse. Diğer durumda, yani cevabı bilmek için  $r$  tane sorunun hepsini sormak isterse kaybeder, bu durumda kazanan Jian-Jia'dır.

Oyunu daha eğlenceli hale getirmek için, Jian-Jia Tayvan'ın gerçek uçuş ağını dikkate almayıp Mei-Yu'nun önceki sorduğu sorularına ve bunlara kendisinin verdiği cevaplara bağlı olarak uçuş ağını oyun süresince istediği gibi değiştirebileceği üzerine anlaşılır. Sizin göreviniz Jian-Jia'nın oyunu kazanması için sorulara hangi cevabı vermesi gerektiğini bulmak.

## Örnekler

Oyunun kurallarını üç örnekle açıklayalım. Her bir örnekte şehir sayısı  $n = 4$  ve soru-cevap raundu sayısı  $r = 6$  dir.

İlk örnekte (aşağıdaki tablo), Jian-Jia kaybeder çünkü 4. raund sonunda Mei-Yu herhangi iki şehir arasında uçuş olduğunu Jian-Jia 5. ve 6. sorulara ne cevap verirse versin kesinlikle bilir.

raund	soru	cevap
1	0, 1	yes
2	3, 0	yes
3	1, 2	no
4	0, 2	yes
-----	-----	-----
5	3, 1	no
6	2, 3	no

İkinci örnekte, Mei-Yu 3. raund sonunda Jian-Jia 4., 5. ve 6. sorulara ne cevap verirse versin 0 ve 1 numaralı şehirler arasında uçuş olamayacağını kesinlikle bilir. Bu yüzden, Jian-Jia yine kaybeder.

raund	soru	cevap
1	0, 3	no
2	2, 0	no
3	0, 1	no
----	-----	-----
4	1, 2	yes
5	1, 3	yes
6	2, 3	yes

Son örnekte, Mei-Yu 6 soru cevaplanmadan herhangi iki şehir arasında uçuş olup olmadığını kesin olarak belirleyemez. Bu yüzden oyunu Jian-Jia kazanır. Aşağıdaki tabloyu detaylı incelersek, Jian-Jia son soruyu *yes* olarak cevapladığından herhangi iki şehir arasında uçuş mümkündür. Buna karşın Jian-Jia son soruyu *no* olarak cevaplasaydı herhangi iki şehir arası uçuş mümkün olmayacaktı.

raund	soru	cevap
1	0, 3	no
2	1, 0	yes
3	0, 2	no
4	3, 1	yes
5	1, 2	no
6	2, 3	yes

## Görev

Jian-Jia'ye oyunu kazandıracak bir program yazın. Mei-Yu ve Jian-Jia'nın birbirinin stratejisini bilmediğine dikkat edin. Mei-Yu şehir çiftlerini herhangi bir sırada sorabilir ve Jian-Jia sonraki soruların ne olacağını bilmeden o anki sorulan soruyu anında cevaplar. Şu iki fonksiyonu yazmalısınız.

- `initialize(n)` -- Öncelikle `initialize` fonksiyonunuz çağrılacaktır. Parametere  $n$  şehir sayısıdır.
- `hasEdge(u, v)` -- Daha sonra `hasEdge` fonksiyonu  $r = n(n - 1)/2$  kez çağrılacaktır. Her bir çağırma Mei-Yu'nun sıradaki sorusunu temsil eder. Siz  $u$  ve  $v$  şehirleri arasında doğrudan uçuş olup olmadığını döndürmelisiniz. Doğrudan uçuş varsa 1, yoksa 0 döndürülmelidir.

## Altgörevler

altgörev	puan	$n$
1	15	$n = 4$
2	27	$4 \leq n \leq 80$
3	58	$4 \leq n \leq 1500$

# Gerçekleştirim detayları

İsmi `game.c`, `game.cpp` ya da `game.pas` olan bir dosya göndermelisiniz. Bu dosya yukarıda tanımlanan fonksiyonları aşağıda verilen yapıda gerçekleştirir.

## C/C++ programları

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

## Pascal programları

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

## Örnek grader

Örnek grader girdi'yi aşağıdaki formatta okur:

- satır 1:  $n$
- takip eden  $r$  satır: her bir satır  $u$  ve  $v$  şehirlerine ilişkin soruyu ifade eden iki tamsayı içerir.