



## Game

Jian-Jia er en ung gutt som elsker å spille spill. Når han blir stilt et spørsmål, foretrekker han å spille spill i stedet for å svare direkte. Jian-Jia møtte vennen sin Mei-Yu og fortalte henne om flynettverket i Taiwan. Det finnes  $n$  byer i Taiwan (nummerert  $0, \dots, n - 1$ ), og noen av de er koblet sammen av flyruter. Hver flyrute kobler sammen to byer og kan brukes i begge retninger.

Mei-Yu spurte Jian-Jia om hvorvidt det er mulig å reise mellom hvilke som helst to byer med fly (enten direkte eller indirekte). Jian-Jia ville ikke avsløre svaret, men foreslo å spille et spill i stedet. Mei-Yu kan stille ham spørsmål på formen "Er byene  $x$  og  $y$  direkte forbundet med en flyrute?", og Jian-Jia vil besvare slike spørsmål umiddelbart. Mei-Yu vil spørre om hvert par av byer eksakt én gang, noe som gir totalt  $r = n(n - 1)/2$  spørsmål. Mei-Yu vinner spillet hvis hun etter å ha fått svarene på de første  $i$  spørsmålene for en  $i < r$  klarer å trekke en slutning om hvorvidt nettverket er sammenhengende eller ikke (dvs. hvorvidt det er mulig å reise direkte eller indirekte mellom ethvert par av byer). Hvis ikke, altså hvis hun trenger alle  $r$  spørsmålene, vinner Jian-Jia.

For at spillet skal være morsommere for Jian-Jia har vennene blitt enige om at han ikke trenger å tenke på det ekte Taiwanske flynettverket, og at han kan finne på nettverket etterhvert som spillet skrider frem, og velge svarene basert på Mei-Yus tidligere spørsmål. Oppgaven din er å hjelpe Jian-Jia med å vinne spillet, ved å avgjøre hvordan han burde besvare spørsmålene.

## Eksempler

Vi forklarer spillreglene med tre eksempler. Hvert eksempel har  $n = 4$  byer og  $r = 6$  runder med spørsmål og svar.

I det første eksempelet (den følgende tabellen) taper Jian-Jia fordi Mei-Yu etter 4. runde vet med sikkerhet at det går an å reise mellom ethvert par av byer med fly, uansett hvordan Jian-Jia besvarer spørsmål 5 eller 6.

Runde	Spørsmål	Svar
1	0, 1	ja
2	3, 0	ja
3	1, 2	nei
4	0, 2	ja
-----	-----	-----
5	3, 1	nei
6	2, 3	nei

I det neste eksempelet kan Mei-Yu etter runde 3 bevise at uansett hvordan Jian-Jia besvarer spørsmål 4, 5 eller 6, kan man *ikke* reise mellom by 0 og 1 med fly, så Jian-Jia taper igjen.

Runde	Spørsmål	Svar
1	0, 3	nei
2	2, 0	nei
3	0, 1	nei
----	-----	-----
4	1, 2	ja
5	1, 3	ja
6	2, 3	ja

I det siste eksempelet kan ikke Mei-Yu avghøre hvorvidt man kan reise mellom ethvert par av byer med fly inntil alle seks spørsmålene er besvart, så Jian-Jia *vinner* spillet. Mer spesifikt, fordi Jian-Jia svarte *ja* på det siste spørsmålet (i den følgende tabellen), er det mulig å reise mellom ethvert par av byer. Men hvis Jian-Jia hadde svart *nei* til det siste spørsmålet, hadde det vært umulig.

Runde	Spørsmål	Svar
1	0, 3	nei
2	1, 0	ja
3	0, 2	nei
4	3, 1	ja
5	1, 2	nei
6	2, 3	ja

## Oppgave

Skriv et program som hjelper Jian-Jia med å med å vinne spillet. Merk at hverken Mei-Yu eller Jian-Jia kjenner til den andres strategi. Mei-Yu kan spørre om bypar i en hvilken som helst rekkefølge, og Jian-Jia må besvare dem umiddelbart uten å vite om de fremtidige spørsmålene. Du må implementere de følgende to funksjonene:

- `initialize(n)` -- Vi vil kalle din `initialize` først. Parameteret  $n$  er antallet byer.
- `hasEdge(u, v)` -- Deretter vil vi kalle `hasEdge`  $r = n(n - 1)/2$  ganger. Disse kallene representerer Mei-Yu sine spørsmål, i den rekkefølgen hun stiller dem. Du må svare hvorvidt det finnes et direktefly mellom by  $u$  og  $v$ . Mer spesifikt skal returverdien være 1 hvis det finnes et direktefly og 0 hvis ikke.

## Deloppgaver

Hver deloppgave består av flere spill. Du vil bare få poeng for en deloppgave hvis programmet ditt vinner alle spillene for Jian-Jia.

Deloppgave	Poeng	$n$
1	15	$n = 4$

Deloppgave	Poeng	$n$
2	27	$4 \leq n \leq 80$
3	58	$4 \leq n \leq 1500$

## Implementasjonsdetaljer

Du må sende inn eksakt én fil, som må hete `game.c`, `game.cpp` eller `game.pas`. Denne filen skal implementere funksjonene som er beskrevet ovenfor med de følgende signaturene:

### C/C++-programmer

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

### Pascal-programmer

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

### Eksempel-grader

Eksempel-graderen leser input på det følgende formatet:

- Linje 1:  $n$
- De følgende  $r$  linjene: hver linje inneholder to heltall  $u$  og  $v$  som beskriver et spørsmål om by  $u$  og  $v$ .