



## Hra (Game)

Ja-n'ko má rád hry a hádanky. Keď sa ho jeho priateľka Mar-i-enka pýtala na súvislosť siete leteckých liniek na Taiwane, vymyslel si namiesto odpovede hru, ktorú si zahrajú.

Na začiatku hry si Mar-i-enka zvolila číslo  $n$ , a následne jej Ja-n'ko oznámil nasledovné informácie: Na Taiwane je  $n$  miest. Tie sú očíslované od 0 po  $n - 1$ . Niektoré dvojice miest sú prepojené priamou leteckou linkou. Každá letecká linka je obojsmerná.

Mar-i-enka sa opýtala Ja-n'ka, či je Taiwan súvislý - teda či sa medzi každou dvojicou miest dá cestovať pomocou vhodnej postupnosti leteckých liniek. Ja-n'ko jej odpovedal, že jej neodpovie. Namiesto toho musí Mar-i-enka odpoveď zistiť hraním hry. Počas hry sa môže Mar-i-enka pýtať Ja-n'ka otázky. Každá otázka musí byť nasledovného typu: "Sú mestá  $x$  a  $y$  spojené *priamou* leteckou linkou?" Ja-n'ko odpovie na otázku okamžite, teda skôr ako Mar-i-enka položí nasledujúcu otázku.

Existuje presne  $r = n(n - 1)/2$  dvojíc miest, a teda  $r$  možných otázok. Mar-i-enka vyhrá hru, ak sa dozvie, či je Taiwan súvislý, po položení menej ako  $r$  otázok. Ja-n'ko vyhrá, ak Mar-i-enku donúti položiť všetkých  $r$  otázok.

Samozrejme, Ja-n'kove odpovede nemusia zodpovedať reálnej mape Taiwanu. Presnejšie, Ja-n'ko si svoje odpovede môže počas hry voliť aké len chce. A ešte presnejšie, Ja-n'kove odpovede bude voliť tvoj program. A musí ich voliť tak, aby úplne každú možnú hru vyhral - bez ohľadu na to, akú stratégiu (teda aké poradie pýtania otázok) použije Mar-i-enka.

## Príklady

Hru budeme ilustrovať tromi príkladmi. V každom príklade máme  $n = 4$  mestá, a teda Mar-i-enka postupne položí  $r = 6$  otázok.

V prvom príklade (uvedenom v nasledujúcej tabuľke) Ja-n'ko *prehral*, lebo po 4. otázke je už Mar-i-enke jasné, že Taiwan musí byť súvislý - a to bez ohľadu na Ja-n'kove odpovede na otázky 5 a 6.

kolo	otázka	odpoveď
1	0, 1	áno
2	3, 0	áno
3	1, 2	nie
4	0, 2	áno
-----	-----	-----
5	3, 1	nie
6	2, 3	nie

V druhom príklade opäť Mar-i-enka vyhrala, a to už po otázke 3. Už v tom okamihu je totiž Mar-i-enke

jasné, že Taiwan nemôže byť súvislý - a to bez ohľadu na Ja-n'kove odpovede na otázky 4, 5 a 6. Presnejšie, nech by Ja-n'ko odpovedal na posledné tri otázky akokoľvek, určite sa už nebude dať cestovať napríklad medzi mestami 0 a 1.

kolo	otázka	odpoveď
1	0, 3	nie
2	2, 0	nie
3	0, 1	nie
-----	-----	-----
4	1, 2	áno
5	1, 3	áno
6	2, 3	áno

V poslednom, tretom príklade Ja-n'ko konečne vyhral. Totiž ani po piatej otázke ešte Mar-i-enke nebolo jasné, či Taiwan je alebo nie je súvislý, a teda musela položiť aj poslednú, šiestu otázku. Všimni si, že keďže Ja-n'ko na šiestu otázku odpovedal "áno", Taiwan je súvislý. Ak by ale odpovedal "nie", Taiwan by súvislý nebol.

kolo	otázka	odpoveď
1	0, 3	nie
2	1, 0	áno
3	0, 2	nie
4	3, 1	áno
5	1, 2	nie
6	2, 3	áno

## Úloha

Napíš program, ktorý bude vyššie popísanú hru hrať namiesto Ja-n'ka a vždy vyhrať. (V každej podúlohe sa bude hrať viacero hier. Ak tvoj program čo len jednu z nich prehrá, dostane za dotyčnú podúlohu 0 bodov.)

Tvoj program pritom nevie, akú stratégiu použije Mar-i-enka (teda testovač) a teda musí fungovať pre všetky možné poradia pýtania sa otázok. Navyše, tvoj program ani počas behu nevie, v akom poradí bude Mar-i-enka klásť otázky v budúcnosti. Na Mar-i-enkine otázky musí tvoj program reagovať *online*, teda každú zodpovedať skôr ako sa dozvie nasledujúcu.

Musíš implementovať nasledujúce dve funkcie:

- `initialize(n)` -- Tvoju funkciu `initialize` zavoláme presne raz, a to hneď na začiatku. Jej volaním oznámime tvojmu programu hodnotu  $n$ , ktorú má použiť.
- `hasEdge(u, v)` -- Následne zavoláme tvoju funkciu `hasEdge` presne  $r$ -krát. Volania tejto funkcie predstavujú Mar-i-enkine otázky, v poradí v akom sa ich pýta. Môžete predpokladať, že na každú dvojicu vrcholov sa Mar-i-enka opýta práve raz. Na každú otázku musíte okamžite odpovedať: dať návratovú hodnotu 1 ak chcete povedať, že mestá  $u$  a  $v$  sú spojené priamou linkou, resp. návratovú hodnotu 0 ak chcete povedať, že priamo spojené nie sú.

# Podúlohy

podúloha	body	$n$
1	15	$n = 4$
2	27	$4 \leq n \leq 80$
3	58	$4 \leq n \leq 1500$

## Detaily implementácie

Odvzdávate presne jeden súbor, nazvaný `game.c`, `game.cpp` alebo `game.pas`. V tomto súbore by mali byť implementované vyššie popísané funkcie. Musia mať nasledujúce hlavičky:

### C/C++

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

### Pascal

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

## Ukážkový grader

Ukážkový grader, ktorý máte k dispozícii, očakáva vstup v nasledovnom formáte:

- riadok 1:  $n$
- každý z nasledujúcich  $r$  riadkov: čísla  $u$  a  $v$  predstavujúce jednu otázku