



Žaidimas

Jaunasis Jian-Jia mėgsta žaisti žaidimus. Paklaustas jis mieliau atsako žaisdamas žaidimą, nei tiesiogiai. Jian-Jia sutiko savo draugę Mei-Yu ir papasakojo jai apie Taivano lėktuvų tinklą. Taivane yra n miestų (sunumeruotų $0, \dots, n-1$), kai kurie iš jų sujungti oro maršrutais. Kiekvienas maršrutas jungia du miestus ir yra dvipusis.

Mei-Yu paklausė Jian-Jia, ar įmanoma lėktuvu nusukti iš bet kurio vieno miesto į bet kurią kitą (tiesiogiai arba per tarpinius miestus). Jian-Jia nenorėjo taip lengvai atsakyti, todėl pasiūlė sužaisti žaidimą.

Mei-Yu gali klausti Jian-Jia "Ar miestus x ir y jungia tiesioginis skrydis?" ir Jian-Jia iš karto atsakys į tokius klausimus. Jei Mei-Yu klaus apie kiekvieną miestų porą lygiai vieną kartą, iš viso užduos $r = n(n-1)/2$ klausimų.

Mei-Yu laimi žaidimą, jei, gavusi atsakymus į pirmuosius i klausimų $i < r$, ji gali pasakyti, ar skrydžių tinklas jungus, t.y. ar įmanoma nusukti tarp visų miestų porų (tiesiogiai arba per tarpinius miestus). Kitaip tariant, jei Mei-Yu reikia užduoti r klausimų, kad gautų atsakymą, laimi Jian-Jia.

Kad žaidimas Jian-Jia būtų smagesnis, draugai nusprendė pamiršti tikrąjį Taivano skrydžių tinklą ir kurti tinklą žaidimo metu, t.y. parinkti atsakymus remiantis ankstesniais Mei-Yu klausimais. Padėkite Jian-Jia laimėti žaidimą nusprenddami, kaip jam atsakinėti į draugės klausimus.

Pavyzdžiai

Paaškinsime žaidimo taisyklės trimis pavyzdžiais. Kiekvienas pavyzdys turi $n = 4$ miestus ir $r = 6$ klausimų/atsakymų raundus.

Pirmajame pavyzdyje (apačioje esančioje lentelėje), Jian-Jia *pralaimi*, nes po 4 raundo Mei-Yu tikrai žino, kad gali iš bet kurio miesto gali nusukti į bet kurią kitą, nesvarbu kaip Jian-Jia atsakys į klausimus 5 ir 6.

raundas	klausimas	atsakymas
1	0, 1	yes
2	3, 0	yes
3	1, 2	no
4	0, 2	yes
-----	-----	-----
5	3, 1	no
6	2, 3	no

Kitame pavyzdyje Mei-Yu po trečiojo raundo gali įrodyti, kad nesvarbu, kaip Jian-Jia atsakys į likusius

4, 5 ir 6 klausimus, *neįmanoma* skraidyti tarp visų miestų, todėl Jian-Jia vėl pralaimi.

raundas	klausimas	atsakymas
1	0, 3	no
2	2, 0	no
3	0, 1	no
----	-----	-----
4	1, 2	yes
5	1, 3	yes
6	2, 3	yes

Paskutiniame pavyzdyje Mei-Yu negali nustatyti, ar įmanoma skraidyti tarp visų miestų kol visi 6 klausimai neatsakyti, todėl Jian-Jia laimi šį žaidimą. Kadangi Jian-Jia atsakė "taip" į paskutinį klausimą (žr. lentelėje toliau), įmanoma iš bet kurio miesto pasiekti bet kurią kitą. Jei būtų atsakęs "ne", būtų neįmanoma.

raundas	klausimas	atsakymas
1	0, 3	no
2	1, 0	yes
3	0, 2	no
4	3, 1	yes
5	1, 2	no
6	2, 3	yes

Užduotis

Parašykite programą, kuri padėtų Jian-Jia laimėti žaidimą. Atkreipiame dėmesį, kad abu žaidėjai nežino vienas kito strategijos. Mei-Yu gali klausti apie skrydžius tarp miestų porų bet kuria tvarka, ir Jian-Jia turi atsakyti iš karto, nežinodamas būsimųjų klausimų.

Parašykite šias funkcijas:

- `initialize(n)` – pirmiausia iškviesime `initialize`. n yra miestų skaičius.
- `hasEdge(u, v)` – tada kviesime `hasEdge` $r = n(n-1)/2$ kartų. Šie kvietimai yra Mei-Yu klausimai nurodyti kvietimo tvarka. Reikia atsakyti, ar yra tiesioginis skrydis tarp miestų u ir v . T.y., reikia grąžinti 1, jei tiesioginis skrydis yra, ir 0, jei nėra.

Dalinės užduotys

Kiekvieną dalinę užduotį sudaro keli žaidimai. Taškai bus skiriami už dalinę užduotį tik tuo atveju, jei programa padės Jian-Jia laimėti visus žaidimus.

dalinė užduotis	taškai	n
1	15	$n = 4$

dalinė užduotis	taškai	n
2	27	$4 \leq n \leq 80$
3	58	$4 \leq n \leq 1500$

Reikalavimai

Įkelkite failą, pavadintą `game.c`, `game.cpp` arba `game.pas`. Jame turi būti aukčiau aprašytos funkcijos.

Programuojantiems C/C++

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

Programuojantiems Paskaliu

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

Pavyzdinis vertintojas

Pavyzdinis vertintojas skaito duomenis šiuo formatu:

- 1-a eilutė: n
- likusios r eilučių: kiekvienoje eilutėje yra po du sveikuosius skaičius u ir v , kurie aprašo klausimą apie miestus u ir v .