



Խաղ

Ջիան-Ջիան փոքրիկ տղա է: Նա շատ է սիրում խաղալ: Երբ նրան հարց են տալիս, նա ուղիղ պատասխան տալու փոխարեն նախընտրում է խաղ խաղալ: Ջիան-Ջիան հանդիպեց իր ընկեր Մեյ-Յունին և նրան պատմեց Թայվանի թռիչքուղիների ցանցի մասին: Թայվանում կա n քաղաք (համարակալված $0, \dots, n - 1$ թվերով): Որոշ քաղաքներ իրար միացված են թռիչքուղիներով: Յուրաքանչյուր թռիչքուղի միացնում է երկու քաղաք: Թռիչքուղիները երկկողմանի են:

Մեյ-Յունն հարցրեց Ջիան-Ջիային - հնարավոր է արդյոք ցանկացած քաղաքից ինքնաթիռով գնալ ցանկացած այլ քաղաք (ուղիղ թռիչքով կամ տրանզիտով): Ջիան-Ջիան պատասխանը տալու փոխարեն առաջարկեց խաղ խաղալ: Մեյ-Յունն կարող է նրան հետևյալ տեսքի հարց տալ. " x և y քաղաքները *ուղիղ* կապ ունե՞ն", և Ջիան-Ջիան այդ հարցերին անմիջապես կպատասխանի: Մեյ-Յունն բոլոր քաղաքների յուրաքանչյուր զույգի վերաբերյալ հարց տալու դեպքում կտա ընդամենը $r = n(n - 1)/2$ հարց: Մեյ-Յունն հաղթում է խաղը, եթե առաջին i հարցերի պատասխանները ստանալուց հետո, որտեղ $i < r$, նա կարող է պարզել թե հնարավոր է արդյոք ցանկացած երկու քաղաքի միջև կապ հաստատել (ուղիղ կամ տրանզիտ): Այլ կերպ ասած, եթե տալիս է բոլոր r հարցերը, ուրեմն հաղթողը Ջիան-Ջիան է:

Որպեսզի Ջիան-Ջիայի համար խաղն ավելի ուրախ լինի, ընկերները պայմանավորվեցին, որ նա կարող է մոռանալ Թայվանի թռիչքուղիների գործող ցանցի մասին, իբրև թե նոր ցանց է պատրաստվում խաղի ընթացքում, և փոխարենը հարցին ընտրում է պատասխան հաշվի առնելով Մեյ-Յունի մինչ այդ տված հարցերը: Ձեր խնդիրն է օգնել Ջիան-Ջիային հաղթելու համար՝ ասելով նրան թե ինչպես պետք է պատասխանի:

Օրինակներ

Բացատրենք խաղի կանոնները երեք օրինակների միջոցով: Բոլոր օրինակներում քաղաքների քանակը $n = 4$ է, հարցերի և պատասխանների փուլերի քանակը $r = 6$ է:

Առաջին օրինակում (հաջորդիվ աղյուսակը), Ջիան-Ջիան *պարտվում է*, որովհետև 4-րդ փուլից հետո Մեյ-Յունն գիտի, որ ցանկացած երկու քաղաքների միջև կապ կա, անկախ նրանից, թե Ջիան-Ջիան ինչպես կպատասխանի 5-րդ և 6-րդ հարցերին:

փուլ	հարց	պատասխան
1	0, 1	այո

փուլ	հարց	պատասխան
2	3, 0	այո
3	1, 2	ոչ
4	0, 2	այո
-----	-----	-----
5	3, 1	ոչ
6	2, 3	ոչ

Հաջորդ օրինակում Մեյ-Յուն կարող է ապացուցել, որ 3-րդ փուլից հետո, անկախ նրանից, թե Ջիան-Ջիան ինչպես կպատասխանի 4-րդ, 5-րդ և 6-րդ հարցերին, *հնարավոր չի լինի* 0 և 1 քաղաքների միջև թռիչքներով կապ հաստատել, այնպես որ Ջիան-Ջիան կրկին պարտվում է:

փուլ	հարց	պատասխան
1	0, 3	ոչ
2	2, 0	ոչ
3	0, 1	ոչ
-----	-----	-----
4	1, 2	այո
5	1, 3	այո
6	2, 3	այո

Վերջին օրինակում մինչև բոլոր վեց հարցերին պատասխան չտրվի, Մեյ-Յուն չի կարող պարզել ցանկացած քաղաքից ցանկացած այլ քաղաք հնարավոր է թռչել, թե ոչ: Այսպիսով, Ջիան-Ջիան *հաղթում է* խաղը: Մասնավորապես, եթե Ջիան-Ջիան վերջին հարցին պատասխանի *այո* (ինչպես հաջորդիվ ներկայացված աղյուսակում է), ապա հնարավոր է ցանկացած քաղաքից թռչել ցանկացած այլ քաղաք: Սակայն, եթե Ջիան-Ջիան վերջին հարցին պատասխաներ *ոչ*, ապա դա հնարավոր չէր լինի:

փուլ	հարց	պատասխան
1	0, 3	ոչ
2	1, 0	այո
3	0, 2	ոչ
4	3, 1	այո
5	1, 2	ոչ
6	2, 3	այո

Խնդիր

Գրեք, իսնդրեմ, ծրագիր, որն օգնում է Ջիան-Ջիային հաղթելու խաղը:

Նկատենք, որ ո՛չ Ջիան-Ջիան, ո՛չ Մեյ-Յուն, մեկը մյուսի ռազմավարության մասին չգիտեն: Մեյ-Յուն կարող քաղաքների զույգերի մասին հարցնել կամայական հերթականությամբ, և Ջիան-Ջիան պետք է հարցերին անմիջապես պատասխանի, առանց իմանալու հետագա հարցերը: Պահանջվում է իրականացնել հետևյալ երկու ֆունկցիաները.

- `initialize(n)` -- Մենք սկզբում կկանչենք Ձեր `initialize` ֆունկցիան: n պարամետրը քաղաքների քանակն է:
- `hasEdge(u, v)` -- Ապա մենք կկանչենք `hasEdge` ֆունկցիան $r = n(n - 1)/2$ անգամ: Այդ կանչերը ներկայացնում են Մեյ-Յունի հարցերը այն հերթականությամբ, ինչ հերթականությամբ նա տալիս է դրանք: Դուք պետք է պատասխանեք u և v քաղաքների միջև ուղիղ թռիչք կա՞, թե՞ ոչ: Մասնավորապես, ուղիղ թռիչքի առկայության դեպքում վերադարձի արժեքը պետք է լինի 1, հակառակ դեպքում պետք է լինի 0:

Ենթախնդիրներ

Ենթախնդիր	Միավոր	n
1	15	$n = 4$
2	27	$4 \leq n \leq 80$
3	58	$4 \leq n \leq 1500$

Իրականացման մանրամասներ

Դուք պետք է ներկայացնեք ճիշտ մեկ ֆայլ, որի անունը պետք է լինի `game.c`, `game.cpp` կամ `game.pas`: Այդ ֆայլում պետք է իրականացնել վերը նկարագրված ֆունկցիաները, օգտագործելով հետևյալ վերնագրերը:

C/C++ ծրագրեր

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

Pascal ծրագրեր

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

Գրեյդերի օրինակ

Գրեյդերի օրինակը ներածում է մուտքային տվյալները հետևյալ ձևաչափով.

- Տող 1: n
- հաջորդ r տողերը: յուրաքանչյուր տող պարունակում է երկու ամբողջ

u և v թվեր, որոնք սկարագրում են u և v քաղաքների վերաբերյալ հարցը: