



Mäng

Jian-Jia on poiss, kellele meeldib mängida. Kui küsida talt küsimus, eelistab ta otsesele vastamisele mängude mängimist. Jian-Jia kohtas oma sõpra Mei-Yu'd ning rääkis talle Taiwani lennuliinide võrgustikust. Taiwanil on n linna (nummerdatud $0, \dots, n - 1$), millest osade vahel käivad lennuliinid. Iga liin ühendab kaht linna ja sellel saab lennata mõlemas suunas.

Mei-Yu küsis Jian-Jia käest, kas igast linnast saab lennata igasse teise linna (kas otse või ümberistumistega). Jian-Jia ei tahtnud vastust avaldada, vaid soovitas selle asemel mängu. Mei-Yu saab talt küsida küsimusi vormis "Kas linnad x ja y on omavahel *otseühenduses*?" ning Jian-Jia vastab sellistele küsimustele kohe. Mei-Yu küsib seda iga linnapaari kohta täpselt ühe korra, saades kokku $r = n(n - 1)/2$ küsimust. Mei-Yu võidab mängu juhul, kui ta saab pärast esimese i vastuse (mingi $i < r$ korral) saamist järeldada, kas võrgustik on sidus, s.t kas iga linnade paari puhul on võimalik esimesest linnast teise lennata või mitte. Vastasel korral, kui tal on vaja kõik r küsimust ära kulutada, võidab mängu Jian-Jia.

Et mäng oleks Jian-Jia jaoks huvitavam, otsustasid sõbrad, et Taiwani tegelik lennuliinide võrgustik pole oluline ja ta võib võrgustiku jooksvalt välja mõelda, valides oma vastuseid Mei-Yu eelmiste küsimuste põhjal. Sinu ülesanne on aidata Jian-Jial mäng võita, otsustades, kuidas ta peaks küsimustele vastama.

Näited

Seletame mängureegleid kolme näitega. Igas näites on $n = 4$ linna ning $r = 6$ küsimuste ja vastuste vooru.

Esimeses näites (vt tabelit) Jian-Jia *kaotab*, sest pärast neljandat vooru teab Mei-Yu kindla peale, et iga linnade paari vahel on võimalik reisida, sõltumata sellest, kuidas Jian-Jia küsimustele 5 ja 6 vastab.

voor	küsimus	vastus
1	0, 1	jah
2	3, 0	jah
3	1, 2	ei
4	0, 2	jah
-----	-----	-----
5	3, 1	ei
6	2, 3	ei

Järgmises näites saab Mei-Yu pärast 3 vooru tõestada, et ükskõik, kuidas Jian-Jia viimastele küsimustele vastab, *ei ole* linnade 0 ja 1 vahel võimalik lennata, nii et Jian-Jia kaotab jälle.

voor	küsimus	vastus
1	0, 3	ei
2	2, 0	ei
3	0, 1	ei
----	-----	-----
4	1, 2	jah
5	1, 3	jah
6	2, 3	jah

Viimases näites ei saa Mei-Yu järeldada, kas igast linnast on võimalik igasse teise linna lennata, kuni kõik kuus küsimust on vastatud, seega Jian-Jia *võidab*. Täpsemalt, kuna Jian-Jia vastas viimasele küsimusele *jah* (vt järgmist tabelit), on iga linnadepaari vahel ühendus olemas. Kui ta oleks aga vastanud *ei*, siis ei oleks.

voor	küsimus	vastus
1	0, 3	ei
2	1, 0	jah
3	0, 2	ei
4	3, 1	jah
5	1, 2	ei
6	2, 3	jah

Ülesanne

Kirjuta programm, mis aitab Jian-Jial mängu võita. Pane tähele, et Mei-Yu ja Jian-Jia ei tea teineteise strateegiat. Mei-Yu võib linnade kohta küsida suvalises järjekorras ja Jian-Jia peab neile kohe vastama, teadmata järgnevate küsimuste järjekorda. Realiseerida tuleb kaks funktsiooni:

- `initialize(n)` -- seda kutsutakse välja esimesena. Parameeter n on linnade arv.
- `hasEdge(u, v)` -- seda kutsutakse välja $r = n(n - 1)/2$ korda. Need väljakutsed tähistavad Mei-Yu küsimusi selles järjekorras, milles ta neid küsib. Neile küsimustele tuleb tagastada 1, kui linnade u ja v vahel on otseühendus, ning 0, kui seda ei ole.

Alamülesanded

Iga alamülesanne koosneb mitmest mängust. Programm saab alamülesande eest punkte ainult siis, kui Jian-Jia võidab kõik mängud.

alamülesanne	punkte	n
1	15	$n = 4$
2	27	$4 \leq n \leq 80$
3	58	$4 \leq n \leq 1500$

Realisatsioon

Esitada tuleb täpselt üks fail nimega `game.c`, `game.cpp` või `game.pas`. Selles failis peavad olema eelpool kirjeldatud alamprogrammid järgmiste signatuuridega:

C ja C++

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

Pascal

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

Näidishindaja

Näidishindaja loeb sisendit järgmises vormingus:

- Esimesel real: arv n .
- Järgmisel r real: igal real on kaks täisarvu u ja v , mis kirjeldavad u ja v vahelise otseühenduse kohta käivat küsimust.