

Datortīkls

Skolas datorklasē ir N datori, kas savā starpā savienoti ar tīkla vadiem. Katrs tīkla vads savieno divus dažādus datorus. Zināms ka jebkuriem diviem datoriem izpildās šāda īpašība, ja datori savā starpā nav savienoti pa tiešo ar vadu, tad vienmēr ziņojumus iespējams nosūtīt izmantojot citus ar tīkla vadiem savienotos datorus, tos lietojot kā starpniekiem. Ziņojums vienmēr tiek sūtīts pa īsāko ceļu: tā lai ziņojums nonāktu pēc iespējas mazāk *starpniekdatoros* (tajos datoros, kas nav ne ziņas sūtītājs, ne saņēmējs).

Ādams un Billijs, kas lieto atšķirīgus datorklases datorus a un b , vēlas noskaidrot īsāko ceļu starp viņu datoriem. Viņi nezina tīkla vadu izvietojumu, bet viņi var sūtīt ziņojumus starp jebkuriem diviem datorklases datoriem un noskaidrot apmeklēto starpniekdatoru skaitu.

Ādams un Billijs nav draugos ar datoru tāpēc viņi lūdz tevi palīgā noskaidrot īsāko ceļu starp viņu datoriem sūtot pēc iespējas maz ziņojumu.

Uzdevums

Atrodiet īsāko ceļu starp datoriem a un b nepārsniedzot atļauto ziņojumu sūtījumu skaitu.

Implementācija

Jums jāimplementē procedūra `findRoute(N, a, b)` ar šādiem parametriem:

- N — datoru skaits datorklasē (tie ir sanumurēti ar skaitļiem no 1 līdz N)
- a, b — Ādama un Billija datoru numuri ($a \neq b$ un $1 \leq a, b \leq N$)

Jūsu procedūra `findRoute` var izsaukt funkciju `ping(i, j)`, kas kā parametrus saņem divu dažādu datoru numurus ($i \neq j$ un $1 \leq i, j \leq N$) un atgriež starpniekdatoru skaitu, kas tika izmantots, lai nosūtītu ziņojumu no datora i uz j .

Jūsu procedūrai `findRoute` jāapraksta īsākais ceļš, pa kuru varētu tikt sūtīts ziņojums no a uz b . Īsākā ceļa aprakstīšana jāveic atkārtoti izsaucot procedūru `travelTo(k)`, kas parametrā saņem tā datora numuru, kas saņems ziņojumu kā nākamais ($1 \leq k \leq N$). Ziņojums sākumā atrodas datorā a , kad tiek izsaukta metode `travelTo(k)` ziņojums tiek pārsūtīts uz datoru k .

Papildus pamatnosacījumiem (laika un atmiņas ierobežojumiem, darbība bez izpildlaika kļūdām) Jūsu iesūtītajam risinājumam, lai tas tiktu ieskaitīts, jāievēro šādi nosacījumi:

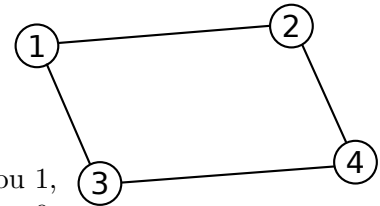
- kad `findRoute` beidz darbu, ziņojumam jābūt nonākušam datorā b ,
- katriem diviem secīgiem datoriem jābūt savienotiem ar tīkla vadu,
- ziņojums jāšūta pa īsāko ceļu,
- funkcijas `ping` izsaukumu skaits M nedrīkst pārsniegt atļauto izsaukumu skaitu (skat. Vērtēšana),
- funkcija `ping` un procedūra `travelTo` jāizsauc tikai ar pieļaujamām parametru vērtībām.

Piemērs

Par piemēru ņemsim diagrammā attēloto datoru tīklu (aplī attēlo datorus, bet līnijas — tīkla vadus). Datortīklā kopā ir $N = 4$ datori. Pieņemsim, ka Ādams un Billijs lieto datorus $a = 1$ and $b = 4$.

Tiks izsaukta procedūra

```
findRoute(4, 1, 4).
```



Šajā gadījumā tās uzvedība varētu būtu šāda:

```
izsauc ping(1, 4), kas atgriež vērtību 1,  
izsauc ping(1, 2), kas atgriež vērtību 0,  
izsauc ping(2, 4), kas atgriež vērtību 0.
```

Ar šo informāciju pietiek, lai noteiktu ka $1 \rightarrow 2 \rightarrow 4$ ir īsākais ceļš 1 uz 4. Šo ceļu var aprakstīt šādi:

```
izsauc travelTo(2),  
izsauc travelTo(4),  
findRoute beidz darbu.
```

Vērtēšana

Visiem apakšuzdevumiem ir spēkā $2 \leq N \leq 1000$.

Apakšuzdevums 1 (25 punkti): starp katriem diviem datoriem ir tieši viens īsākais ceļš; M Nedrīkst pārsniegt $2N$.

Apakšuzdevums 2 (25 punkti): M Nedrīkst pārsniegt N^2 .

Apakšuzdevums 3 (25 punkti): M Nedrīkst pārsniegt $4N$.

Apakšuzdevums 4 (25 punkti): M Nedrīkst pārsniegt $2N$.

Ierobežojumi

Laika ierobežojums: 1 s.

Atmiņas ierobežojums: 64 MB.

Eksperimentēšana

Pārbaudes programma uz Jūsu datora lasīs ievaddatus no standarta ievada. Pirmajā ievaddatu rindā jābūt četriem veseliem skaitļiem N, a, b un M ierobežojumam. Katrā no nākamajām N rindām jābūt N veseliem skaitļiem, kas apraksta tīkla vadu izvietojumu: i -tās rindas j -tais



skaitlis ($i \neq j$) norāda starpniekdatoru skaitu kādu izmantotu, ja sūtītu ziņu no i uz j . Nav svarīgi kādas ir norādītās vērtības ja $i = j$.

Šeit redzams, kā būtu aprakstīts iepriekšaplūkotais piemērs, ja M ierobežots ar 100:

```
4 1 4 100
0 0 0 1
0 0 1 0
0 1 0 0
1 0 0 0
```