

Computer Network

Vidurinės mokyklos kompiuterių klasėje N kompiuterių yra kabeliais sujungti į bendrą tinklą. Kiekvienas kabelis jungia du skirtingus kompiuterius. Kai kurie kompiuteriai gali būti nesusjungti kabeliu tiesiogiai, tačiau tarp bet kurių dviejų kompiuterių galima persiųsti pranešimą tiesiogiai arba per tarpinius kabeliais sujungtus kompiuterius. Pranešimai visada pasirenka trumpiausią kelią, t.y. pakeliui aplanko mažiausią galimą skaičių kompiuterių (neskaičiuojant pranešimą siunčiančio ir gaunančio kompiuterių).

Adam ir Billy, kurie klasėje naudojami skirtingais kompiuteriais a ir b , norėtų rasti trumpiausią kelią tarp savo kompiuterių. Jie nežino, kaip išvedžioti kabeliai, bet gali siųsti pranešimus tarp kiekvienos kompiuterių poros ir suskaičiuoti, kiek tarpinių kompiuterių siunčiami pranešimai aplanko.

Adam ir Billy nelabai gaudosi kompiuteriuose, tad prašo jūsų pagalbos, kad netektų siųsti per daug pranešimų.

Užduotis

Raskite trumpiausią kelią tarp kompiuterių a ir b siųsdami ne daugiau negu leistiną skaičių pranešimų.

Realizacija

Jums reikia realizuoti vieną procedūrą **findRoute**(N , a , b), kurios argumentai yra:

- N — klasėje esančių kompiuterių skaičius (jie sunumeruoti nuo 1 iki N)
- a, b — Adam ir Billy kompiuterių numeriai ($a \neq b$ ir $1 \leq a, b \leq N$)

Jūsų procedūra **findRoute** gali kviesti funkciją **ping**(i , j), kurios argumentai yra dviejų skirtingų kompiuterių numeriai ($i \neq j$ ir $1 \leq i, j \leq N$), o grąžinama reikšmė — tarpinių kompiuterių skaičius pranešimui keliaujant iš kompiuterio i į kompiuterį j .

Jūsų procedūra **findRoute** turi įvardinti trumpiausią kelią, kuriuo kelias pranešimas, išsiųstas iš kompiuterio a į kompiuterį b . Tai turi būti atliekama pakartotinai kviečiant procedūrą **travelTo**(k) su vieninteliu argumentu — kompiuterio, į kurį turi keliauti pranešimas kitu ėjimu ($1 \leq k \leq N$), numeriu. Pranešimas kelionę pradeda kompiuteryje a ir, kai tik iškviečiama procedūra **travelTo**(k), jis persikelia į kompiuterį k .

Papildomai prie standartinių reikalavimų (laiko ir atminties ribojimai, jokių vykdymo klaidų), jūsų sprendimas turėtų tenkinti šiuos papildomus reikalavimus:

- procedūrai **findRoute** baigus darbą pranešimas turėtų būti kompiuteryje b ,
- pranešimo kelyje visi gretimi kompiuteriai privalo būti tiesiogiai sujungti kabeliu,
- tai turi būti trumpiausias galimas kelias,
- funkcijos **ping** iškvietyčių skaičius neturi viršyti M (žr. skyrių Vertinimas),

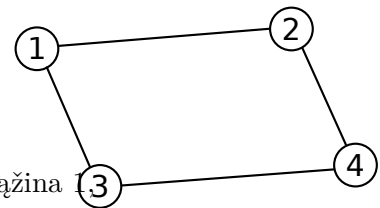
- funkcija `ping` ir procedūra `travelTo` privalo būti kviečiamos tik su leistinomis argumentų reikšmėmis.

Pavyzdys

Tarkime, kad duotas kompiuterių tinklas atrodo taip, kaip pavaizduotas diagromoje žemiau (taškai žymi kompiuterius, o atkarpos — kabelius). Iš viso tinkle yra $N = 4$ kompiuteriai, o Adam ir Billy naudojami kompiuteriais $a = 1$ ir $b = 4$.

Pirmiausiai bus iškviesta procedūra

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



Vienas galimas jos elgesys galėtų būti toks::

kviečiama funkcija `ping(1, 4)`, kuri grąžina 1,
kviečiama funkcija `ping(1, 2)`, kuri grąžina 0,
kviečiama funkcija `ping(2, 4)`, kuri grąžina 0.

Šios informacijos pakanka, kad rastume trumpiausią kelią $1 \rightarrow 2 \rightarrow 4$ iš kompiuterio 1 į kompiuterį 4. Rastas kelias turėtų būti įvardintas tokiu būdu:

kviečiama procedūra `travelTo(2)`,
kviečiama procedūra `travelTo(4)`,
procedūra `findRoute` baigia darbą.

Vertinimas

Visose užduočių grupėse galioja apribojimas $2 \leq N \leq 1000$.

Subtask 1 (25 points): tarp bet kurių dviejų kompiuterių yra lygiai vienas trumpiausias kelias; M negali viršyti $2N$.

Subtask 2 (25 points): M negali viršyti N^2 .

Subtask 3 (25 points): M negali viršyti $4N$.

Subtask 4 (25 points): M negali viršyti $2N$.

Ribojimai

Time limit: 1 s.

Memory limit: 64 MB.

Experimentation

Jūsų kompiuteryje esantis pavyzdinis vertintojas nuskaitys duomenis iš standartinės įvesties. Pirmoje eilutėje turėtų būti keturi sveikieji skaičiai N, a, b, M . Kitose N eilučių turėtų būti po N sveikųjų skaičių, apibūdinančių kabelių išdėstymą: j -asis skaičius i -ojoje eilutėje ($i \neq j$) žymi tarpinių kompiuterių, kurį aplankys pranešimas keliaudamas iš kompiuterio i į kompiuterį j , skaičių. Jeigu $i = j$, gali būti nurodytos bet kokios reikšmės.

Žemiau pateikta įvestis nusako anksčiau pateiktą pavyzdį, kai iškviemų skaičius M apribotas iki 100:

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