



Teemaksud

Jaapani linnad on ühendatud kiirteedega. Kiirteede võrk koosneb N linnast ja M kiirteest. Iga kiirtee ühendab kaht erinevat linna. Iga kahe linna vahel on ülimalt üks kiirtee. Linnad on nummerdatud $0 \dots N - 1$ ja kiirteed $0 \dots M - 1$. Kõik teed on kahesuunalised ja on teada, et igast linnast pääseb neid mööda igasse teise.

Iga kiirtee kasutamise eest tuleb tasuda teemaks, mis sõltub tee **liiklustihedusest**. Liiklus teel võib olla kas **hõre** või **tihe**. Hõreda liikluse korral on teemaks A , tiheda liikluse korral B jeeni. On teada, et $A < B$ ning A ja B väärtused on Sulle antud.

Sul on masin, millele Sa saad anda teede liiklustihedused ja mis leiab nende põhjal minimaalse linnast S linna T (kus $S \neq T$) sõitmise teemaksude summa.

See masin on aga kõigest prototüüp, milles S ja T väärtusi ei saa muuta (need on masina ehituses fikseeritud). Enamgi veel, S ja T väärtused pole Sulle teada. Aga Sa tahaks neid teada. Selleks on Sul vaja anda masinale ette erinevaid andmeid teede liiklustiheduste kohta ja kasutada väljastatud sõidukulused, et nende põhjal tuletada S ja T väärtused. Kuna liiklustiheduste andmete sisestamine kulutab aega, ei taha Sa masinat palju kordi kasutada.

Realisatsioon

Lahendusena tuleb realiseerida üks alamprogramm

```
find_pair(int N, int[] U, int[] V, int A, int B)
```

- N : linnade arv.
- U ja V : massiivid pikkusega M , kus M on kiirteede arv. Tee i ($0 \leq i \leq M - 1$) ühendab linnu $U[i]$ ja $V[i]$.
- A : teemaks hõreda liikluse korral.
- B : teemaks tiheda liikluse korral.
- Seda alamprogrammi kutsutakse igas testis välja täpselt üks kord.
- Pane tähele, et M on massiivide pikkus ja selle saab kätte lehel "Realisatsioon" kirjeldatud viisidel.

Alamprogramm `find_pair` võib kasutada funktsiooni

```
int64 ask(int[] w)
```

- Massiiv w kirjeldab teede liiklustihedusi ja selle pikkus peab olema M .
- Tee i ($0 \leq i \leq M - 1$) liiklustihedust näitab $w[i]$, mille väärtus peab olema kas 0 või 1.
 - $w[i] = 0$ tähendab, et liiklus teel i on hõre.
 - $w[i] = 1$ tähendab, et liiklus teel i on tihe.
- See funktsioon tagastab minimaalse teedemaksude summa, mis kulub linnast S linna T sõitmiseks massiivis w kirjeldatud liiklustiheduste korral.
- Seda funktsiooni võib igas testis välja kutsuda maksimaalselt 100 korda.

Alamprogramm `find_pair` peab vastuse teatamiseks kutsuma välja

```
answer(int s, int t)
```

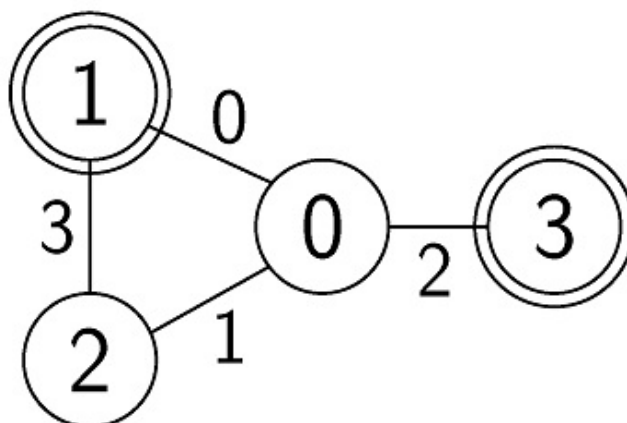
- s ja t peavad olema S ja T (nende järjekord pole oluline).
- Seda alamprogrammi peab välja kutsuma täpselt ühe korra.

Kui Sinu lahendus mõnda eeltoodud tingimustest rikub, saab Su lahendus diagnoosi **Wrong Answer**. Vastasel juhul on diagnoos **Accepted** ja skoor arvutatakse funktsiooni ask väljakutsete arvu põhjal (vt Alamülesanded).

Näide

Olgu $N = 4$, $M = 4$, $U = [0, 0, 0, 1]$, $V = [1, 2, 3, 2]$, $A = 1$, $B = 3$, $S = 1$, $T = 3$.

Keskkond kutsub välja `find_pair(4, [0, 0, 0, 1], [1, 2, 3, 2], 1, 3)`.



Joon number i eeloleval joonisel vastab kiirteele number i .

Mõned võimalikud ask kutsed ja nende tagastatavad väärtused:

Kutse	Väärtus
ask([0, 0, 0, 0])	2
ask([0, 1, 1, 0])	4
ask([1, 0, 1, 0])	5
ask([1, 1, 1, 1])	6

Väljakutsel ask([0, 0, 0, 0]) on liiklus kõigil teedel hõre ja kõigi teede maks seega 1. Odavaim marsruut linnast $S = 1$ linna $T = 3$ on sel juhul $1 \rightarrow 0 \rightarrow 3$. Selle marsruudi teemaksude summa on 2 ja funktsioon ask tagastabki väärtuse 2.

Õige vastuse teatamiseks peab find_pair kutsuma välja kas answer(1, 3) või answer(3, 1).

Abimaterjalide ZIP-arhiivis olev fail sample-01-in.txt vastab sellele näitele. Arhiivis on ka teisi sisendi näiteid.

Piirangud

- $2 \leq N \leq 90\,000$
- $1 \leq M \leq 130\,000$
- $1 \leq A < B \leq 1\,000\,000\,000$
- Iga $0 \leq i \leq M - 1$ korral
 - $0 \leq U[i] \leq N - 1$
 - $0 \leq V[i] \leq N - 1$
 - $U[i] \neq V[i]$
- Iga $0 \leq i < j \leq M - 1$ korral $(U[i], V[i]) \neq (U[j], V[j])$ ja $(U[i], V[i]) \neq (V[j], U[j])$
- Igast linnast pääseb kiirteid mööda igasse teise
- $0 \leq S \leq N - 1$
- $0 \leq T \leq N - 1$
- $S \neq T$

Hindamisprogramm ei ole selles ülesandes adaptiivne. See tähendab, et S ja T pannakse testi alguses paika ja ei sõltu Sinu lahenduse funktsiooniväljakutsetest.

Alamülesanded

1. (5 punkti) kas $S = 0$ või $T = 0$, $N \leq 100$, $M = N - 1$
2. (7 punkti) kas $S = 0$ või $T = 0$, $M = N - 1$
3. (6 punkti) $M = N - 1$, $U[i] = i$, $V[i] = i + 1$ ($0 \leq i \leq M - 1$)
4. (33 punkti) $M = N - 1$
5. (18 punkti) $A = 1$, $B = 2$
6. (31 punkti) Lisapiirangud puuduvad

Kui Su programm saab diagnoosi **Accepted** ja kutsub funktsiooni ask välja X korda, siis arvutatakse Sinu skoor iga testi eest järgmiselt:

- Alamülesanne 1: $P = 5$.
- Alamülesanne 2: Kui $X \leq 60$, siis $P = 7$. Muidu $P = 0$.
- Alamülesanne 3: Kui $X \leq 60$, siis $P = 6$. Muidu $P = 0$.
- Alamülesanne 4: Kui $X \leq 60$, siis $P = 33$. Muidu $P = 0$.
- Alamülesanne 5: Kui $X \leq 52$, siis $P = 18$. Muidu $P = 0$.
- Alamülesanne 6:
 - Kui $X \leq 50$, siis $P = 31$.
 - Kui $51 \leq X \leq 52$, siis $P = 21$.
 - Kui $53 \leq X$, siis $P = 0$.

Pane tähele, et Sinu skoor iga alamülesande eest on selle alamülesande testide skooride miinimum.

Hindamisprogramm

Arhiivis olev hindamisprogramm loeb sisendit järgmises vormingus:

- rida 1: $N M A B S T$
- rida $2 + i$ ($0 \leq i \leq M - 1$): $U[i] V[i]$

Kui Su programm saab diagnoosi **Accepted**, siis kuvab hindamisprogramm Accepted: q, kus q on funktsiooni ask väljakutsete arv.

Kui Su programm saab diagnoosi **Wrong Answer**, siis kuvatakse Wrong Answer: MSG, kus MSG tähendus on järgmine:

- answered not exactly once: Alamprogrammi answer kutsuti välja vähem või rohkem kui üks kord.
- w is invalid: Funktsioonile ask antud w pikkus ei ole M või mõne $w[i]$ ($0 \leq i \leq M - 1$) väärtus pole ei 0 ega 1.
- more than 100 calls to ask: Funktsiooni ask kutsuti välja rohkem kui 100 korda.
- {s, t} is wrong: Alamprogrammi answer kutsuti välja valede s ja t väärtustega.