



Greitkelių mokesčiai

Japonijoje N miestų jungia M greitkelių. Kiekvienas greitkelis jungia du skirtingus miestus. Jokie du greitkeliai nejungia tų pačių dviejų miestų. Miestai sunumeruoti nuo 0 iki $N - 1$, o greitkeliai sunumeruoti nuo 0 iki $M - 1$. Kiekvienu greitkeliu galima važiuoti abiem kryptimis. Greitkeliais galima nuvažiuoti iš bet kurio miesto į bet kurią kitą miestą.

Už važiavimą kiekvienu greitkeliu reikia susimokėti mokestį. Konkretaus greitkelio mokestis priklauso nuo **eismo** sąlygų tame greitkelyje. Eismas greitkelyje būna **mažo intensyvumo** arba **didelio intensyvumo**. Kai eismo intensyvumas mažas, mokestis yra A jenų (Japonijos valiuta). Kai didelis -- B jenų. Garantuojama, kad $A < B$. Atkreipkite dėmesį, kad A ir B reikšmės žinomos.

Turite mašiną, kuri, nurodžius eismo sąlygas visuose greitkeliuose, apskaičiuoja, kiek mažiausiai reikia sumokėti, norint nuvažiuoti iš miesto S į miestą T ($S \neq T$).

Tačiau ši mašina yra tik prototipas. S ir T reikšmės yra fiksuotos (t.y., įprogramuotos mašinos gaminimo metu ir nekeičiamos) ir jums nežinomos. Norite rasti S ir T , todėl ketinate pateikti mašinai kelis skirtingus eismo sąlygų aprašymus, ir panaudoti jos gražinamas kelionės kainas, kad nustatytumėte S ir T reikšmes. Kadangi kiekviena užklausa kainuoja, nenorite mašinos naudoti pernelyg daug kartų.

Realizacija

Parašykite tokią procedūrą:

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

- N : miestų skaičius.
- U ir V : ilgio M masyvai, čia M yra miestus jungiančių greitkelių skaičius. Kiekvienam i ($0 \leq i \leq M - 1$), greitkelis i jungia miestus $U[i]$ ir $V[i]$.
- A : vieno greitkelio mokestis, kai eismo intensyvumas mažas.
- B : vieno greitkelio mokestis, kai eismo intensyvumas didelis.
- Kiekvienam testui ši procedūra iškviečiama lygiai vieną kartą.
- Atkreipkite dėmesį, kad M yra masyvų ilgis, ir jį galima rasti taip, kaip nurodyta pastabose apie realizaciją.

Procedūra `find_pair` gali kviesti šią funkciją:

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

- Masyvas w aprašo eismo sąlygas ir jo ilgis turi būti M .
- Kiekvienam i ($0 \leq i \leq M - 1$), $w[i]$ nurodo eismo sąlygas i -ajame greitkelyje. $w[i]$ reikšmė turi būti lygi 0 arba 1.
 - $w[i] = 0$ reiškia, kad eismo intensyvumas i -ajame greitkelyje mažas.
 - $w[i] = 1$ reiškia, kad eismo intensyvumas i -ajame greitkelyje didelis.
- Ši funkcija grąžina mažiausią kiekį pinigų, kurių reikia sumokėti greitkelių mokesčiams, važiuojant iš miesto S į miestą T , esant eismo sąlygoms, nurodytoms masyve w .
- Šią funkciją galima kviesti daugiausia 100 kartų (kiekvienam testui).

Norėdama pateikti atsakymą, `find_pair` turi iškviesti šią procedūrą:

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

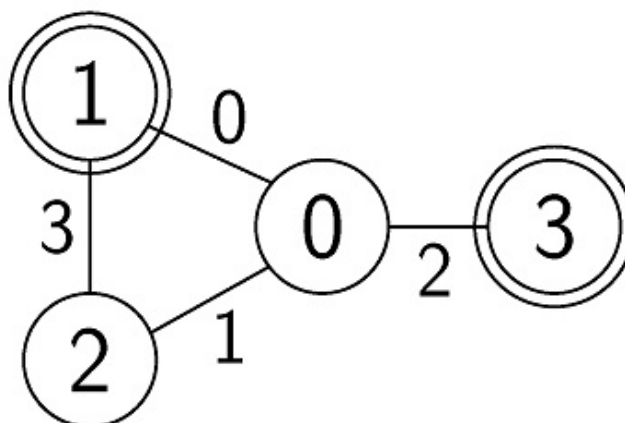
- s ir t turi būti lygūs, atitinkamai, arba S ir T , arba T ir S (tvarka nesvarbi).
- Šią procedūrą reikia iškviesti lygiai vieną kartą.

Jei jūsų programa kurios nors iš šių sąlygų netenkins, ji bus įvertinta **Wrong Answer**. Priešingu atveju, jūsų programa bus įvertinta **Accepted** ir gaunami taškai priklausys nuo to, kiek kartų kviečiama funkcija `ask` (žr. Dalinės užduotys).

Pavyzdys

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

Vertintojas iškviečia `find_pair(4, [0, 0, 0, 1], [1, 2, 3, 2], 1, 3)`.



Šiame paveikslėlyje, skaičiumi i pažymėta briauna atitinka i -ąjį greitkeli. Žemiau pateikta keletas galimų `ask` iškvietimų ir, atitinkamai, `ask` grąžinamos reikšmės:

Iškvietimas	Grąžinama reikšmė
ask([0, 0, 0, 0])	2
ask([0, 1, 1, 0])	4
ask([1, 0, 1, 0])	5
ask([1, 1, 1, 1])	6

Iškviečiama ask([0, 0, 0, 0]). Šiuo atveju, eismas visuose greitkeluose yra mažo intensyvumo ir kiekvieno greitkelio mokestis lygus 1. Pigiausias kelias iš $S = 1$ į $T = 3$ yra $1 \rightarrow 0 \rightarrow 3$. Pilna šios kelionės kaina yra 2. Todėl funkcija grąžina 2.

Kad pateiktų teisingą atsakymą, find_pair turėtų iškviešti answer(1, 3) arba answer(3, 1).

Failas sample-01-in.txt, pateiktas zip formatu suformatuotame pakete, atitinka šį pavyzdį. Kitų pavyzdinių testų įvestys taip pat pateikiamos šiame pakete.

Ribojimai

- $2 \leq N \leq 90\,000$
- $1 \leq M \leq 130\,000$
- $1 \leq A < B \leq 1\,000\,000\,000$
- Kiekvienam $0 \leq i \leq M - 1$
 - $0 \leq U[i] \leq N - 1$
 - $0 \leq V[i] \leq N - 1$
 - $U[i] \neq V[i]$
- $(U[i], V[i]) \neq (U[j], V[j])$ ir $(U[i], V[i]) \neq (V[j], U[j])$ ($0 \leq i < j \leq M - 1$)
- Greitkeliais galima iš bet kurio miesto nuvykti į bet kurį kitą miestą.
- $0 \leq S \leq N - 1$
- $0 \leq T \leq N - 1$
- $S \neq T$

Šiame uždavinyje vertinimo programa NĖRA adaptvyvi. Tai reiškia, kad S ir T reikšmės fiksuojamos, kai vertinimo programa pradeda vykdyti, ir nepriklauso nuo jūsų programos pateikiamų užklausų.

Dalinės užduotys

1. (5 taškai) Vienas iš S ir T yra lygus 0, $N \leq 100$, $M = N - 1$
2. (7 taškai) Vienas iš S ir T yra lygus 0, $M = N - 1$
3. (6 taškai) $M = N - 1$, $U[i] = i$, $V[i] = i + 1$ ($0 \leq i \leq M - 1$)
4. (33 taškai) $M = N - 1$
5. (18 taškų) $A = 1$, $B = 2$

6. (31 taškas) Papildomų ribojimų nėra.

Tarkime, jūsų programa įvertinta **Accepted**, ir X kartų iškviečia funkciją `ask`. Tuomet už šį testą skiriamų taškų skaičius P priklauso nuo to, kuriai dalinei užduočiai jis priklauso. P skaičiuojamas taip:

- Dalinė užduotis nr. 1. $P = 5$.
- Dalinė užduotis nr. 2. Jei $X \leq 60$, $P = 7$. Priešingu atveju, $P = 0$.
- Dalinė užduotis nr. 3. Jei $X \leq 60$, $P = 6$. Priešingu atveju, $P = 0$.
- Dalinė užduotis nr. 4. Jei $X \leq 60$, $P = 33$. Priešingu atveju, $P = 0$.
- Dalinė užduotis nr. 5. Jei $X \leq 52$, $P = 18$. Priešingu atveju, $P = 0$.
- Dalinė užduotis nr. 6.
 - Jei $X \leq 50$, $P = 31$.
 - Jei $51 \leq X \leq 52$, $P = 21$.
 - Jei $53 \leq X$, $P = 0$.

Atkreipkite dėmesį, kad už dalinę užduotį skiriamų taškų skaičius yra lygus minimaliam taškų skaičiui, gautam už tos dalinės užduoties testus.

Pavyzdinė vertinimo programa

Pavyzdinė vertinimo programa skaito duomenis tokiu formatu:

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

Jei jūsų programa įvertinama **Accepted**, pavyzdinė vertinimo programa išveda `Accepted`: q , čia q yra funkcijos `ask` iškvietimų skaičius.

Jei jūsų programa įvertinama **Wrong Answer**, pavyzdinis vertintojas išveda `Wrong Answer`: MSG , čia MSG yra viena iš:

- `answered not exactly once`: Procedūra `answer` buvo iškviesta ne lygiai vieną kartą.
- `w is invalid`: w ilgis, pateiktas funkcijai `ask`, nėra M , arba $w[i]$ nėra lygus nei 0, nei 1, kažkuriam i ($0 \leq i \leq M - 1$).
- `more than 100 calls to ask`: Funkcija `ask` kviečiama daugiau nei 100 kartų.
- `{s, t} is wrong`: Procedūra `answer` kviečiama su neteisinga skaičių pora s ir t .