



Cestnine

Na Japonskem so ceste med seboj povezane z omrežjem avtocest. Omrežje sestavlja N mest in M avtocest. Vsaka avtocesta povezuje par različnih mest. Nobeni dve avtocesti ne povezujeta istega para mest. Mesta so oštevilčena od 0 do $N - 1$. Po vsaki avtocesti se je moč voziti v obe smeri. Z vožnjo po avtocestah je iz poljubnega mesta moč pripotovati v poljubno drugo mesto.

Za vožnjo po vsaki avtocesti se zaračuna cestnina. Znesek cestnine je odvisen od gostote **prometa** na posamezni avtocesti. Promet je bodisi **reddek** bodisi **gost**. Ko je promet reddek, cestnina znaša A jenov (japonska valuta). Ko je promet gost, cestnina znaša B jenov. Zagotovljeno je, da velja $A < B$. Upoštevaj, da sta vrednosti A in B znani.

Imaš stroj, ki ob podanih prometnih razmerah vseh avtocest izračuna najnižji skupni znesek cestnin za potovanje med mestoma S in T ($S \neq T$).

Vendar pa je ta stroj zgolj prototip. Vrednosti S in T sta fiksirani (t.j. "zapečeni" v stroj) in ti nista znani. Tvoja naloga je ugotoviti S in T . Da to dosežeš, lahko stroju podaš več opisov prometnih razmer in na podlagi odgovorov (skupnih vrednosti cestnin) določiš S in T . Ker je podajanje prometnih razmer drago oz. zamudno, si želiš stroj uporabiti čim manjkrat.

Podrobnosti implementacije

Implementiranj naslednjo proceduro:

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

- N : število mest.
- U and V : polji dolžine M , kjer je M število avtocest. Za vsak i ($0 \leq i \leq M - 1$) avtocesta i povezuje mesti $U[i]$ in $V[i]$.
- A : cestnina za avtocesto, ko je promet reddek.
- B : cestnina za avtocesto, ko je promet gost.
- To proceduro se pokliče natanko enkrat za vsak testni primer.
- Upoštevaj, da je vrednost M dolžina polj in da jo lahko pridobiš na način, opisan v implementacijskih opombah.

Procedura `find_pair` lahko kliče naslednjo funkcijo:

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

- Dolžina w mora biti M . Polje w opisuje prometne razmere.
- Za vsak i ($0 \leq i \leq M - 1$) podaja $w[i]$ prometne razmere na avtocesti i . Vrednost $w[i]$ mora biti bodisi 0 bodisi 1.
 - $w[i] = 0$ pomeni, da je promet na avtocesti i redek.
 - $w[i] = 1$ pomeni, da je promet na avtocesti i gost.
- Funkcija vrne najmanjšo skupno vrednost cestnin za potovanje med mestoma S in T pri prometnih razmerah, podanih z w .
- To funkcijo se lahko pokliče največ 100-krat (za vsak testni primer).

Procedura `find_pair` mora poklicati naslednjo proceduro, s katero poda odgovor:

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

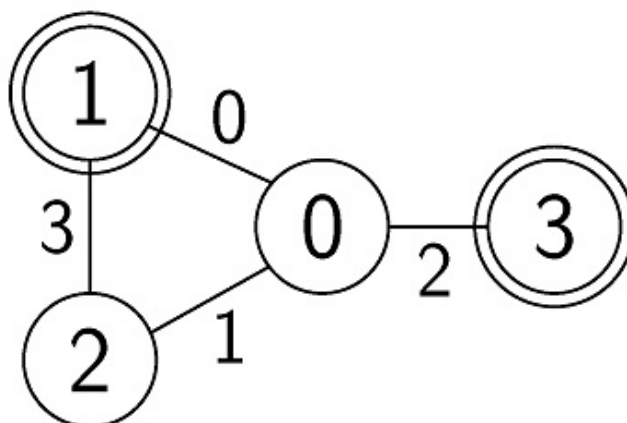
- s in t morata biti par S in T (vrstni red ni pomemben).
- To proceduro se mora poklicati natanko enkrat.

Če kateri izmed zgornjih pogojev ni izpolnjen, bo tvoj program ocenjen z **Wrong Answer**. Sicer bo program ocenjen z **Accepted**, število točk pa se izračuna na podlagi števila klicev funkcije `ask` (glej Podnaloge).

Primer

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

Ocenjevalnik pokliče `find_pair(4, [0, 0, 0, 1], [1, 2, 3, 2], 1, 3)`.



Na zgornji sliki povezava s številko i ustreza avtocesti i . Sledi nekaj možnih klicev funkcije `ask` in ustrezni odgovori:

Klic	Odgovor
ask([0, 0, 0, 0])	2
ask([0, 1, 1, 0])	4
ask([1, 0, 1, 0])	5
ask([1, 1, 1, 1])	6

Ob klicu funkcije `ask([0, 0, 0, 0])` je promet na vsaki avtocesti redek, cestnina za vsako avtocesto pa znaša 1. Najcenejša pot med $S = 1$ in $T = 3$ je $1 \rightarrow 0 \rightarrow 3$. Skupna vrednost cestnin za to pot je 2. Zatorej funkcija vrne 2.

Za pravilen odgovor mora procedura `find_pair` poklicati `answer(1, 3)` ali `answer(3, 1)`.

Datoteka `sample-01-in.txt` v priponki `zip` ustreza temu primeru. Drugi primeri vhoda in izhoda so tudi na voljo v priponki.

Omejitve

- $2 \leq N \leq 90\,000$
- $1 \leq M \leq 130\,000$
- $1 \leq A < B \leq 1\,000\,000\,000$
- Za vsak $0 \leq i \leq M - 1$ velja:
 - $0 \leq U[i] \leq N - 1$
 - $0 \leq V[i] \leq N - 1$
 - $U[i] \neq V[i]$
- $(U[i], V[i]) \neq (U[j], V[j])$ in $(U[i], V[i]) \neq (V[j], U[j])$ ($0 \leq i < j \leq M - 1$)
- Iz poljubnega mesta je preko avtocest moč pripotovati v poljubno drugo mesto.
- $0 \leq S \leq N - 1$
- $0 \leq T \leq N - 1$
- $S \neq T$

Pri tej nalogi ocenjevalnik NI prilagodljiv. To pomeni, da sta S in T vnaprej fiksirana ob zagonu ocenjevalnika in tako nista odvisna od poizvedb, ki jih opravi tvoja rešitev.

Podnaloge

1. (5 točk) ena izmed vrednosti S in T je 0, $N \leq 100$, $M = N - 1$.
2. (7 točk) ena izmed vrednosti S in T je 0, $M = N - 1$.
3. (6 točk) $M = N - 1$, $U[i] = i$, $V[i] = i + 1$ ($0 \leq i \leq M - 1$).
4. (33 točk) $M = N - 1$.
5. (18 točk) $A = 1$, $B = 2$.
6. (31 točk) Brez dodatnih omejitev.

Predpostavi, da je tvoj program ocenjen kot **Accepted** in da izvede X klicev funkcije `ask`. Število točk P za testni primer se glede na številko podnaloge izračuna takole:

- Podnaloga 1. $P = 5$.
- Podnaloga 2. Če velja $X \leq 60$, potem $P = 7$. Sicer $P = 0$.
- Podnaloga 3. Če velja $X \leq 60$, potem $P = 6$. Sicer $P = 0$.
- Podnaloga 4. Če velja $X \leq 60$, potem $P = 33$. Sicer $P = 0$.
- Podnaloga 5. Če velja $X \leq 52$, potem $P = 18$. Sicer $P = 0$.
- Podnaloga 6.
 - Če velja $X \leq 50$, potem $P = 31$.
 - Če velja $51 \leq X \leq 52$, potem $P = 21$.
 - Če velja $53 \leq X$, potem $P = 0$.

Opomba: tvoj rezultat za vsako podnalogo je minimum rezultatov vseh testnih primerov posamezne podnaloge.

Vzorčni ocenjevalnik

Vzorčni ocenjevalnik bere vhod naslednjega formata:

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

Če je tvoj program ocenjen kot **Accepted**, vzorčni ocenjevalnik izpiše `Accepted: q`, kjer je q število klicev funkcije `ask`.

Če je tvoj program ocenjen z **Wrong Answer**, izpiše `Wrong Answer: MSG`, kjer je niz `MSG` eden izmed:

- `answered not exactly once`: Procedura `answer` ni bila poklicana natanko enkrat.
- `w is invalid`: Dolžina w , podana funkciji `ask`, ni M , ali pa vrednost $w[i]$ za nek i ($0 \leq i \leq M - 1$) ni niti 0 niti 1.
- `more than 100 calls to ask`: Funkcija `ask` je poklicana več kot 100-krat.
- `{s, t} is wrong`: Procedura `answer` je poklicana z napačnim parom s in t .