



Highway Tolls

Japanska cestovna mreža se sastoji od N gradova i M cesta. Svaka cesta povezuje dva različita grada, i svaka dva grada su povezana najviše jednom cestom. Gradovi su označeni brojevima od 0 do $N - 1$, a ceste brojevima od 0 do $M - 1$. Svaka cesta je dvosmjerna i postoji put između svaka dva grada.

Za svaku pojedinu cestu se mora platiti cestarina kada se njome putuje. Cestarina neke ceste nije uvijek ista već ovisi o **gustoći prometa** na toj cesti. U svakom trenutku je na pojedinoj cesti promet ili **rijedak** ili **gust**. Kada je promet rijedak, cestarina je A jena. Kada je promet gust, cestarina je B jena. Naravno, vrijedi da je $A < B$, a A i B su fiksne i svima poznate vrijednosti.

JAK (Japanski Auto Klub) upravo provodi uzbudljivu nagradnu igru. Javno su objavili detaljan plan cestovne mreže, te su odabrali dva različita **tajna** grada S i T . Putem web stranice JAK-a, korisnik može specificirati stanje na svim cestama, a web stranica će izračunati i prikazati najmanju ukupnu cijenu cestarine koju je potrebno platiti kako bi se (kada je baš takvo stanje na cestama) otputovalo od tajnog grada S do tajnog grada T .

Vaš cilj je da odredite tajne gradove S i T . Za tu svrhu možete koristiti gore opisanu web stranicu te nekoliko puta specificirati stanje na cestama i upotrijebiti dobivene odgovore o cijeni najjeftinijeg puta između S i T kako bi zaključili o kojim gradovima se radi. Kako je usluga *roaming*-a prilično skupa, ne želite postaviti previše upita.

Implementacijski detalji

Morate implementirati sljedeću funkciju:

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

- N : broj gradova.
- U i V : nizovi duljine M , gdje je M broj cesta. Za svaki i ($0 \leq i \leq M - 1$), cesta i povezuje gradove $U[i]$ i $V[i]$.
- A : cestarina za svaku pojedinu cestu na kojoj je promet rijedak.
- B : cestarina za svaku pojedinu cestu na kojoj je promet gust.
- Ova funkcija se poziva točno jednom za svaki test podatak.
- Primijetite da vrijednost M odgovara duljini polja i može se odrediti shodno dokumentu o implementacijskim uputama.

Vaš program može pozivati sljedeću funkciju:

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

- Duljina niza w mora biti M . Niz w opisuje stanje na cestama.
- Za svaki i ($0 \leq i \leq M - 1$), $w[i]$ opisuje stanje na cesti i . Vrijednost $w[i]$ mora biti 0 ili 1.
 - $w[i] = 0$ znači da je promet na cesti i rijedak.
 - $w[i] = 1$ znači da je promet na cesti i gust.
- Funkcija vraća najmanju moguću ukupnu cestarinu za put od tajnog grada S do tajnog grada T za stanje na cestama opisano zadanim nizom w .
- Ovu funkciju smijete pozvati najviše 100 puta (za pojedini test podatak).

Vaš program mora pozvati sljedeću funkciju kada pronade tajne gradove:

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

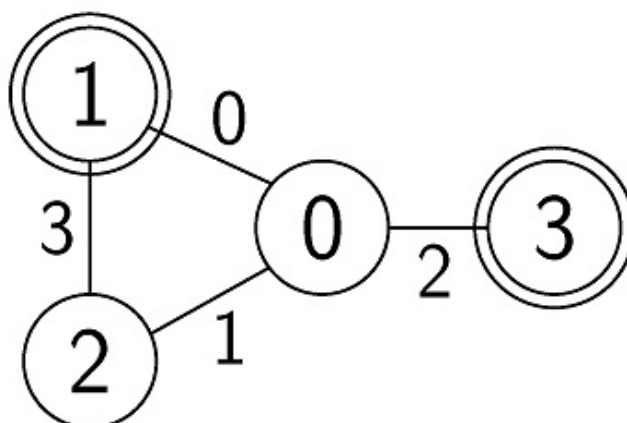
- s i t moraju biti tajni gradovi S i T (redoslijed je nebitan).
- Ovu funkciju morate pozvati točno jednom.

Ako neki od gornjih uvjeta nije zadovoljen, vaš program će biti ocijenjen s rezultatom **Wrong Answer**. Inače, vaš program će biti ocijenjen s rezultatom **Accepted** te će broj bodova ovisiti o broju poziva funkcije `ask` (vidi poglavlje Podzadaci).

Primjer

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

Program `grader` poziva `find_pair(4, [0, 0, 0, 1], [1, 2, 3, 2], 1, 3)`.



Na gornjoj slici, brid označen brojem i odgovara i -toj cesti.

Jedan mogući niz poziva funkcije `ask` je sljedeći:

Poziv	Rezultat
ask([0, 0, 0, 0])	2
ask([0, 1, 1, 0])	4
ask([1, 0, 1, 0])	5
ask([1, 1, 1, 1])	6

U prvom pozivu -- ask([0, 0, 0, 0]) -- promet na svakoj cesti je rijedak i svaka cestarina je, dakle, 1. Najjeftiniji put od grada $S = 1$ do grada $T = 3$ je $1 \rightarrow 0 \rightarrow 3$. Ukupna cestarina za ovaj put je 2. Dakle, funkcija vraća vrijednost 2.

Da bi odgovor bio točan, vaš program mora pozvati `answer(1, 3)` ili `answer(3, 1)`.

Ulazna datoteka `sample-01-in.txt` u privitku odgovara ovom primjeru. Privitak sadrži i druge primjere ulaznih datoteka.

Ograničenja

- $2 \leq N \leq 90\,000$
- $1 \leq M \leq 130\,000$
- $1 \leq A < B \leq 1\,000\,000\,000$
- Za svaki $0 \leq i \leq M - 1$
 - $0 \leq U[i] \leq N - 1$
 - $0 \leq V[i] \leq N - 1$
 - $U[i] \neq V[i]$
- $(U[i], V[i]) \neq (U[j], V[j])$ i $(U[i], V[i]) \neq (V[j], U[j])$ ($0 \leq i < j \leq M - 1$)
- Graf je povezan -- postoji put između svakog para gradova.
- $0 \leq S \leq N - 1$
- $0 \leq T \leq N - 1$
- $S \neq T$

U ovom zadatku, *grader* koji se koristi prilikom evaluacije rješenja NIJE adaptivan. Dakle, gradovi S i T su fiksirani prilikom pokretanja *grader*-a te ne ovise o upitima vašeg rješenja.

Podzadaci

1. (5 bodova) jedan od gradova S i T je grad 0, dodatno vrijedi $N \leq 100$, $M = N - 1$
2. (7 bodova) jedan od gradova S i T je grad 0, dodatno vrijedi $M = N - 1$
3. (6 bodova) $M = N - 1$, $U[i] = i$, $V[i] = i + 1$ ($0 \leq i \leq M - 1$)
4. (33 boda) $M = N - 1$
5. (18 bodova) $A = 1$, $B = 2$
6. (31 bod) Nema dodatnih ograničenja.

Pretpostavimo da je za neki test podatak vaš program ocijenjen s rezultatom **Accepted**, te da je pritom pozvao funkciju ask X puta. Vaši bodovi P za ovaj test podatak ovise o podzadatku i o broju X na sljedeći način:

- Podzadatak 1. $P = 5$.
- Podzadatak 2. Ako je $X \leq 60$, $P = 7$. Inače $P = 0$.
- Podzadatak 3. Ako je $X \leq 60$, $P = 6$. Inače $P = 0$.
- Podzadatak 4. Ako je $X \leq 60$, $P = 33$. Inače $P = 0$.
- Podzadatak 5. Ako je $X \leq 52$, $P = 18$. Inače $P = 0$.
- Podzadatak 6.
 - Ako je $X \leq 50$, $P = 31$.
 - Ako je $51 \leq X \leq 52$, $P = 21$.
 - Ako je $53 \leq X$, $P = 0$.

Broj bodova za svaki podzadatak je najmanji osvojeni broj bodova među svim test podacima u tom podzadatku.

Ogledni *grader*

Ogledni *grader* očekuje ulazne podatke u sljedećem formatu:

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

Ako je vaš program ocijenjen s rezultatom **Accepted**, ogledni *grader* ispisuje Accepted: q, gdje je q broj poziva funkcije ask.

Ako je vaš program ocijenjen s rezultatom **Wrong Answer**, ogledni *grader* ispisuje Wrong Answer: MSG, gdje je MSG jedno od sljedećeg:

- answered not exactly once: Funkcija answer nije pozvana točno jednom.
- w is invalid: Duljina niza w prilikom poziva funkcije ask nije jednaka M ili $w[i]$ nije niti 0 niti 1 za neki i ($0 \leq i \leq M - 1$).
- more than 100 calls to ask: Funkcija ask je pozvana više od 100 puta.
- {s, t} is wrong: Funkcija answer je pozvana s krivim gradovima s i t.