



Autópálya díj

Japánban N várost M kétirányú autópálya köt össze. Minden autópálya két különböző várost köt össze és bármely két város között legfeljebb 1 van. A városokat 0-tól $N - 1$ -ig, az autópályákat 0-tól $M - 1$ -ig sorszámozzuk. Bármely városból bármely városba el lehet jutni autópályákon.

Minden autópálya használatért forgalomtól függő díjat kell fizetni. A forgalom **gyenge** vagy **erős** lehet. **Gyenge** forgalomnál az autópályadíj A yen, **erős** forgalomnál pedig B . Biztosan teljesül, hogy $A < B$.

Van egy géped, ami az adott forgalmat figyelembe véve kiszámítja az S és T város közötti utazás minimális költségét ($S \neq T$).

Az S és T a gépbe van beépítve, amit nem ismersz. Ezeket kell meghatároznod!

Beállíthatod minden autópályára az ottani forgalmat és lekérdezheted, hogy mi az S és T közötti utazás költsége ezzel a forgalommal számolva.

Megvalósítás

A következő függvényt kell megvalósítanod.

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

- N : a városok száma
- U és V : M elemű vektorok, ahol M az autópályák száma. Az i . autópálya az $U[i]$ és $V[i]$ városokat köti össze ($0 \leq i \leq M - 1$).
- A : a gyenge forgalom esetén fizetendő díj
- B : az erős fordgalom esetén fizetendő díj
- Ezt a függvényt pontosan egyszer hívják tesztetesként.
- Az M értéke lekérdezhető, mint a vektorok hossza.

A `find_pair` a következő függvényt hívhatja:

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

- w tartalmazza a forgalom leírását, M hosszúságú kell legyen!
- $w[i]$ az i . autópálya forgalmát adja meg, értéke 0 (gyenge forgalom) vagy 1 (erős

forgalom) lehet ($0 \leq i \leq M - 1$).

- A visszaadott függvényérték az S és T közötti legolcsóbb utazás költsége, ha a forgalmat a w írja le.
- Legfeljebb 100-szor hívhatod.

A következő függvénnyel kell megadnod az eredményt:

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

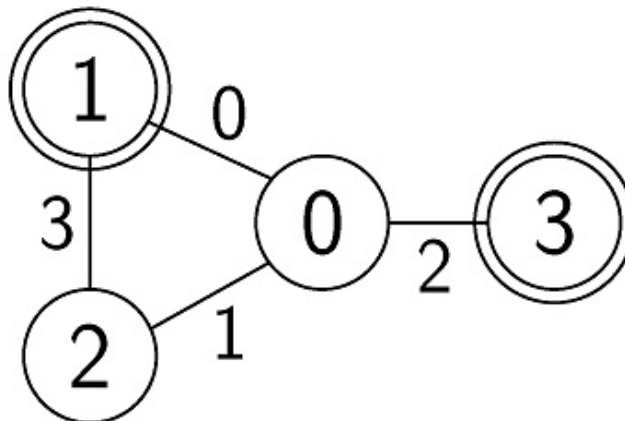
- s és t a kitalálendő két város sorszáma, tetszőleges sorrendben.
- Ezt a függvényt egyszer kell meghívnod!

Ha a fenti feltételek nem teljesülnek, **Wrong Answer** értékelést kapsz, egyébként **Accepted** lesz az értékelés és a pontszámod az ask hívások számától függ a részfeladatoknál leírtak szerint.

Példa

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

Az értékelő hívása: `find_pair(4, [0, 0, 0, 1], [1, 2, 3, 2], 1, 3)`.



Néhány lehetséges ask hívás és eredménye:

Hívás	Eredmény
<code>ask([0, 0, 0, 0])</code>	2
<code>ask([0, 1, 1, 0])</code>	4
<code>ask([1, 0, 1, 0])</code>	5
<code>ask([1, 1, 1, 1])</code>	6

Az `ask([0, 0, 0, 0])` hívásban minden forgalom gyenge és minden díj $A = 1$. A

legolcsóbb út $S = 1$ és $T = 3$ között $1 \rightarrow 0 \rightarrow 3$. Az utazás költsége 2. Így a függvényed értéke 2.

Tehát a `find_pair` függvényben az `answer(1, 3)` vagy `answer(3, 1)` hívással kell megadnod az eredményt!

A tömörített mintában a `sample-01-in.txt` tartalmazza ezt a példát. Más példák is vannak benne.

Korlátok

- $2 \leq N \leq 90\,000$
- $1 \leq M \leq 130\,000$
- $1 \leq A < B \leq 1\,000\,000\,000$
- Minden $0 \leq i \leq M - 1$ -re
 - $0 \leq U[i] \leq N - 1$
 - $0 \leq V[i] \leq N - 1$
 - $U[i] \neq V[i]$
- $(U[i], V[i]) \neq (U[j], V[j])$ és $(U[i], V[i]) \neq (V[j], U[j])$ ($0 \leq i < j \leq M - 1$)
- Bármely városból bármely városba el lehet jutni autópályákon.
- $0 \leq T \leq N - 1$
- $S \neq T$

Az értékelő nem adaptív, azaz S és T értéke rögzített, válasza nem függ attól, hogy mit kérdeztél korábban..

Részfeladatok

1. (5 pont) S vagy T értéke 0, $N \leq 100$, $M = N - 1$
2. (7 pont) S vagy T értéke 0, $M = N - 1$
3. (6 pont) $M = N - 1$, $U[i] = i$, $V[i] = i + 1$ ($0 \leq i \leq M - 1$)
4. (33 pont) $M = N - 1$
5. (18 pont) $A = 1$, $B = 2$
6. (31 pont) nincs további feltétel

Ha az értékelés **Accepted**, és X ask hívásod volt, akkor a P pontszámod az alábbiak szerint számolják:

- Az 1. részfeladatban $P = 5$.
- A 2. részfeladatban ha $X \leq 60$, akkor $P = 7$, egyébként $P = 0$.
- A 3. részfeladatban ha $X \leq 60$, akkor $P = 6$, egyébként $P = 0$.
- A 4. részfeladatban ha $X \leq 60$, akkor $P = 33$, egyébként $P = 0$.
- Az 5. részfeladatban ha $X \leq 52$, akkor $P = 18$, egyébként $P = 0$.
- A 6. részfeladatban
 - ha $X \leq 50$, akkor $P = 31$.

- ha $51 \leq X \leq 52$, akkor $P = 21$.
- ha $53 \leq X$, akkor $P = 0$.

Minden részfeladatra a tesztesetek pontszámának minimumát kapod.

Minta értékelő

A bemenetet az alábbi formában olvassa:

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

Ha az értékelés **Accepted**, akkor a mintaértékelő az Accepted: q üzenetet írja ki, ahol q az ask hívások száma.

Wrong Answer esetén az üzenet Wrong Answer: MSG, ahol MSG az alábbiak valamelyike lehet:

- answered not exactly once: Az answer-t nem egyszer hívtad.
- w is invalid: A w hossza az ask hívásban nem M vagy $w[i]$ nem 0 és nem 1 ($0 \leq i \leq M - 1$).
- more than 100 calls to ask: 100-nál több ask hívás.
- {s, t} is wrong: Az answer függvényt nem a helyes s és t paraméterekkel hívtad.