



Dálniční mýta

Japonská města jsou propojena sítí dálnic. Tato síť se sestává z N měst a M dálnic. Každá dálnice propojuje právě dvě města. Žádné dvě dálnice nepropojují stejnou dvojici měst. Města jsou očíslována od 0 do $N - 1$, dálnice jsou očíslované od 0 do $M - 1$. Všechny dálnice jsou obousměrné. Z každého města se lze pomocí dálnic dostat do libovolného jiného města.

Na každé dálnici se vybírá mýto. Velikost mýta závisí na tom, jaká je na dálnici hustota dopravy - ta je buď **nízká**, nebo **vysoká**. Pokud je hustota dopravy nízká, mýto je A yenů, zatímco je-li hustota dopravy vysoká, mýto je B yenů. Platí, že $A < B$. Hodnoty A a B vám jsou známy.

Pořídili jste si přístroj, který na základě hustoty dopravy na všech dálnicích dokáže spočítat nejmenší možný poplatek, který je potřeba zaplatit za cestu z města S do města T ($S \neq T$).

Přístroj je zatím ve fázi testování. Hodnoty S a T jsou fixní (zadrátované v přístroji) a vy je bohužel neznáte. Vaším úkolem je tyto hodnoty určit. Abyste to udělali, zadáte několikrát přístroji, jaká je hustota dopravy na dálnicích, a podle toho, jaké dostanete odpovědi, určíte S a T . Každé použití přístroje je nicméně drahé, takže jej nechcete používat zbytečně mnohokrát.

Implementační detaily

Vaším úkolem je implementovat následující proceduru:

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

- N : počet měst.
- U a V : pole délky M , kde M je celkový počet dálnic. Pro každé i ($0 \leq i \leq M - 1$) platí, že dálnice i propojuje města $U[i]$ a $V[i]$.
- A : velikost mýta, pokud je hustota dopravy nízká.
- B : velikost mýta, pokud je hustota dopravy vysoká.
- Tato procedura je pro každý testovací vstup volána právě jednou.
- Hodnotu M lze získat jako délku polí, jak to udělat naleznete na papíru *Poznámky k implementaci*.

Procedura `find_pair` může volat následující funkci:

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

- Délka pole w musí být M . Pole w udává, jaká je hustota dopravy na všech dálnicích.
- Pro každé i ($0 \leq i \leq M - 1$) hodnota $w[i]$ udává, jaká je hustota dopravy na dálnici i . Hodnota $w[i]$ je vždy 0, nebo 1.
 - $w[i] = 0$ znamená, že hustota dopravy na dálnici i je nízká.
 - $w[i] = 1$ znamená, že hustota dopravy na dálnici i je vysoká.
- Tato funkce vrátí nejmenší možný poplatek, který je třeba zaplatit, chceme-li podniknout cestu města S do města T , přičemž hustota dopravy je dána polem w .
- Pro každý testovací vstup je maximální možný počet volání této funkce 100.

`find_pair` nakonec vrátí odpověď pomocí následující funkce:

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

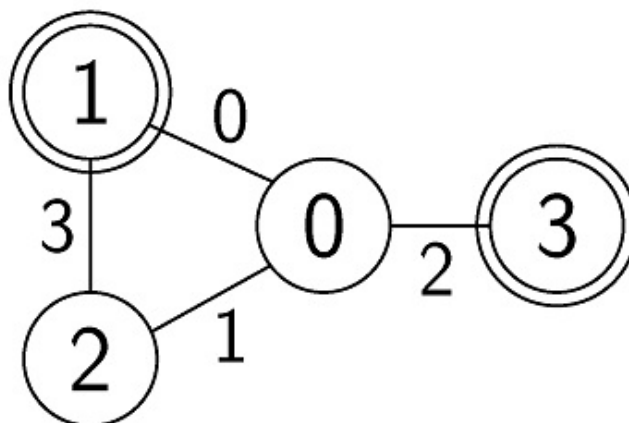
- s a t musí být dvě města S a T (na pořadí nezáleží).
- Tato procedura musí být zavolána právě jednou.

Pokud některá ze zmíněných podmínek není splněná, váš program bude vyhodnocen jako **Wrong Answer**. Jinak bude váš program vyhodnocen jako **Accepted** a počet získaných bodů bude vypočten na základě počtu volání funkce `ask` způsobem popsáním v sekci Podúlohy.

Příklad

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

Vyhodnocovač zavolá `find_pair(4, [0, 0, 0, 1], [1, 2, 3, 2], 1, 3)`.



Hrana označená na obrázku číslem i odpovídá dálnici číslo i . Některé možnosti, jak volat funkci `ask`, a odpovídající vrácené hodnoty této funkce jsou v následující tabulce:

volání	vrácená hodnota
ask([0, 0, 0, 0])	2
ask([0, 1, 1, 0])	4
ask([1, 0, 1, 0])	5
ask([1, 1, 1, 1])	6

Při volání ask([0, 0, 0, 0]) je hustota dopravy každé dálnice nízká a mýto je pokaždé 1. Nejlevnější cesta z $S = 1$ do $T = 3$ je $1 \rightarrow 0 \rightarrow 3$. Celkový poplatek za tuto cestu je 2, funkce proto vrátí 2.

Pro vrácení správné odpovědi má procedura find_pair zavolat buď answer(1, 3), nebo answer(3, 1).

Soubor sample-01-in.txt v zazipovaném archívu odpovídá tomuto příkladu. Archív obsahuje i jiné vzorové vstupy.

Omezení

- $2 \leq N \leq 90\,000$
- $1 \leq M \leq 130\,000$
- $1 \leq A < B \leq 1\,000\,000\,000$
- Pro každé $0 \leq i \leq M - 1$
 - $0 \leq U[i] \leq N - 1$
 - $0 \leq V[i] \leq N - 1$
 - $U[i] \neq V[i]$
- $(U[i], V[i]) \neq (U[j], V[j])$ a $(U[i], V[i]) \neq (V[j], U[j])$ ($0 \leq i < j \leq M - 1$)
- Z každého města se lze pomocí dálnic dostat do libovolného jiného města.
- $0 \leq S \leq N - 1$
- $0 \leq T \leq N - 1$
- $S \neq T$

Vyhodnocovač v tomto případě NENÍ adaptivní. To znamená, že S a T jsou zafixovány na začátku běhu vyhodnocovače a nezávisí na tom, co dělá vaše řešení.

Podúlohy

1. (5 bodů) Jedno město z dvojice S a T je 0, $N \leq 100$, $M = N - 1$
2. (7 bodů) Jedno město z dvojice S a T je 0, $M = N - 1$
3. (6 bodů) $M = N - 1$, $U[i] = i$, $V[i] = i + 1$ ($0 \leq i \leq M - 1$)
4. (33 bodů) $M = N - 1$
5. (18 bodů) $A = 1$, $B = 2$
6. (31 bodů) Žádná další omezení

Předpokládejme, že váš program je vyhodnocen jako **Accepted** a počet volání funkce `ask` je X . Počet získaných bodů P za tento testovací vstup se pak v závislosti na podúloze spočte následujícím způsobem:

- Podúloha 1. $P = 5$.
- Podúloha 2. Jestliže $X \leq 60$, pak $P = 7$. Jinak $P = 0$.
- Podúloha 3. Jestliže $X \leq 60$, pak $P = 6$. Jinak $P = 0$.
- Podúloha 4. Jestliže $X \leq 60$, pak $P = 33$. Jinak $P = 0$.
- Podúloha 5. Jestliže $X \leq 52$, pak $P = 18$. Jinak $P = 0$.
- Podúloha 6.
 - Jestliže $X \leq 50$, pak $P = 31$.
 - Jestliže $51 \leq X \leq 52$, pak $P = 21$.
 - Jestliže $53 \leq X$, pak $P = 0$.

Počet bodů získaných za podúlohu je vypočten jako minimum z počtu bodů za odpovídající testovací vstupy.

Ukázkový testovač

Ukázkový testovač čte vstup v následujícím formátu:

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

Je-li váš program vyhodnocen jako **Accepted**, ukázkový testovač vytiskne `Accepted: q`, kde q je počet volání funkce `ask`.

Je-li váš program vyhodnocen jako **Wrong Answer**, testovač vytiskne `Wrong Answer: MSG`, kde `MSG` je jedna z následujících zpráv:

- `answered not exactly once`: Procedura `answer` nebyla zavolána právě jednou.
- `w is invalid`: Délka pole `w` zadaného funkci `ask` jako parametr není M , nebo pro nějaké i ($0 \leq i \leq M - 1$) není hodnota `w[i]` rovna ani 0, ani 1.
- `more than 100 calls to ask`: Počet volání funkce `ask` přesáhl 100.
- `{s, t} is wrong`: Procedura `answer` byla zavolána s nesprávnou dvojicí `s` a `t`.