



Mýto na diaľniciach

V Japonsku je N miest očíslovaných od 0 po $N - 1$. Niektoré dvojice miest sú spojené obojsmernými diaľnicami. Diaľnic je M a sú očíslované od 0 po $M - 1$. Každá diaľnica spája dve rôzne mestá. Každá dvojica miest je spojená najviac jednou priamou diaľnicou a platí, že v tejto diaľničnej sieti sa dá cestovať medzi ľubovoľnou dvojicou miest.

Za prejazd diaľnicou sa platí mýto. Sú len dve možné ceny mýta, spoločné pre všetky diaľnice:

- ak je dotyčná diaľnica práve mimo špičky (light traffic), platí sa za prejazd ňou A jenov.
- ak je na diaľnici špička (heavy traffic), platí sa B jenov, pričom $A < B$.

Hodnoty A a B poznáš.

Máš v rukách prototyp stroja, ktorý má počítať najlacnejšiu cestu z mesta S do mesta T ($S \neq T$) v závislosti od dopravných podmienok. Cena cesty je pochopiteľne rovná súčtu cien za prejazd jednotlivými diaľnicami, ktoré ju tvoria.

Keďže však ide o prototyp, mestá S a T nevieš zmeniť: sú natvrdo zadané v hardvéri stroja. Jediné, čo vieš so strojom robiť, je zadať mu konkrétny popis dopravnej situácie a opýtať sa ho na najlacnejšiu cenu cesty z S do T .

Hodnoty S a T nepoznáš. Pomocou dostatočne malého počtu otázok vyššie popísaného typu zisti, ktoré dve mestá sú S a T .

Implementačné detaily

Tvojou úlohou je naprogramovať nasledovnú funkciu:

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

- N je počet miest
- U a V sú polia dĺžky M popisujúce jednotlivé diaľnice. Pre každé i od 0 po $M - 1$ platí, že diaľnica i spája mestá $U[i]$ a $V[i]$.
- A je mýto pre ľubovoľnú diaľnicu mimo špičky
- B je mýto pre ľubovoľnú diaľnicu počas špičky
- Testovač zavolá túto tvoju funkciu presne raz pre každý vstup.

- Hodnota M nie je zadaná explicitne. Vieš ju získať spôsobom popísaným v dokumente nazvanom Implementačný nieľad.

Tvoja funkcia `find_pair` smie opakovane volať nasledujúcu funkciu gradera:

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

- Pole w popisuje dopravné podmienky. Toto pole musí mať presne M prvkov.
- Pre každé i od 0 po $M - 1$ platí, že hodnota $w[i]$ popisuje stav dopravy na diaľnici číslo i , a to nasledovne:
 - $w[i] = 0$ znamená, že na diaľnici i **nie je špička**
 - $w[i] = 1$ znamená, že na diaľnici i **je špička**
- Táto funkcia ti vráti najmenšiu možnú celkovú cenu cesty z mesta S do mesta T za predpokladu, že dopravná situácia zodpovedá tebou zadanému polu w .
- Pre každý vstup platí, že túto funkciu smieš zavolať nanajvýš 100-krát.

Keď už zistíš správnu dvojicu miest S a T , tvoj program ju musí oznámiť testovaču, a to tak, že tvoja funkcia `find_pair` zavolá nasledujúcu funkciu:

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

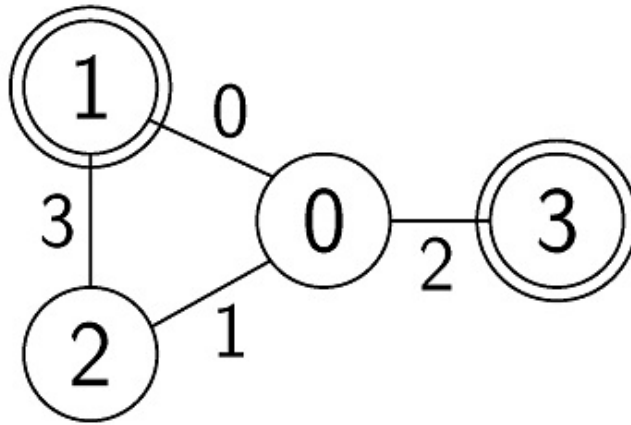
- s a t musia byť čísla miest S a T (v ľubovoľnom poradí).
- Túto funkciu musíš zavolať práve raz.

Ak porušíš niektoré z vyššie popísaných podmienok, dostaneš verdikt **Wrong Answer**. Ak ich všetky dodržíš, dostaneš verdikt **Accepted** a možno aj nejaké body. Popis toho, kedy koľko bodov dostaneš, nájdeš v časti Podúlohy.

Príklad

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

Pre tento vstup testovač zavolá tvoju funkciu `find_pair(4, [0, 0, 0, 1], [1, 2, 3, 2], 1, 3)`.



V obrázku vyššie označujú čísla pri hranách čísla príslušných diaľníc. Nasleduje jedna možná postupnosť volaní funkcie ask a hodnôt, ktoré vráti:

Ty zavolaš	Dostaneš návratovú hodnotu
ask([0, 0, 0, 0])	2
ask([0, 1, 1, 0])	4
ask([1, 0, 1, 0])	5
ask([1, 1, 1, 1])	6

Pri volaní ask([0, 0, 0, 0]) nie je nikde dopravná špička, za každú diaľnicu sa teda platí $A = 1$ jen. Najlacnejšia cesta z $S = 1$ do $T = 3$ je $1 \rightarrow 0 \rightarrow 3$. Táto cesta stojí 2 jeny, funkcia ask teda vráti hodnotu 2.

Keď tvoja funkcia find_pair určí správne hodnoty S a T , musí práve raz zavolať answer(1, 3) alebo answer(3, 1).

V prílohe k tejto úlohe nájdete súbor sample-01-in.txt ktorý zodpovedá tomuto príkladu, a tiež ďalšie príklady vstupu.

Obmedzenia

- $2 \leq N \leq 90\,000$
- $1 \leq M \leq 130\,000$
- $1 \leq A < B \leq 1\,000\,000\,000$
- Pre každé i od 0 po $M - 1$ vrátane:
 - $0 \leq U[i] \leq N - 1$
 - $0 \leq V[i] \leq N - 1$
 - $U[i] \neq V[i]$
- Pre každé i a j od 0 po $M - 1$, pričom $i < j$:
 - $(U[i], V[i]) \neq (U[j], V[j])$ a $(U[i], V[i]) \neq (V[j], U[j])$

- Dialničná sieť je zaručene súvislá.
- $0 \leq S \leq N - 1$
- $0 \leq T \leq N - 1$
- $S \neq T$

Grader v tejto úlohe **nie je adaptívny**. V každom teste sú mestá S a T zvolené vopred a nezávisia od toho, aké volania funkcie ask spraví tvoj program.

Podúlohy

1. (5 bodov) jedno z S a T je 0, $N \leq 100$, $M = N - 1$
2. (7 bodov) jedno z S a T je 0, $M = N - 1$
3. (6 bodov) $M = N - 1$ a pre všetky i od 0 po $M - 1$ platí $U[i] = i$ a $V[i] = i + 1$
4. (33 bodov) $M = N - 1$
5. (18 bodov) $A = 1$, $B = 2$
6. (31 bodov) bez ďalších obmedzení

Za predpokladu, že tvoj program dostane v danom teste verdikt **Accepted**, počet bodov P za tento test dostaneš tak, že sa pozrieš na podúlohu, ktorú práve riešiš, a na číslo X : počet volaní funkcie ask, ktoré tvoj program potreboval.

- Podúloha 1. Bez ohľadu na X dostaneš $P = 5$ bodov.
- Podúloha 2. Ak $X \leq 60$, dostaneš $P = 7$ bodov, inak dostaneš $P = 0$ bodov.
- Podúloha 3. Ak $X \leq 60$, dostaneš $P = 6$ bodov, inak dostaneš $P = 0$ bodov.
- Podúloha 4. Ak $X \leq 60$, dostaneš $P = 33$ bodov, inak dostaneš $P = 0$ bodov.
- Podúloha 5. Ak $X \leq 52$, dostaneš $P = 18$ bodov, inak dostaneš $P = 0$ bodov.
- Podúloha 6.
 - Ak $X \leq 50$, dostaneš plných $P = 31$ bodov.
 - Ak $51 \leq X \leq 52$, dostaneš ako cenu útechy aspoň $P = 21$ bodov.
 - Ak $53 \leq X$, si chudák a máš $P = 0$ bodov.

Ako už určite čakáš, výsledný počet bodov za každú **podúlohu** dostaneš tak, že zoberieš **minimum** z počtov bodov za jednotlivé testy ktoré patria do tejto podúlohy.

Sample grader

Sample grader očakáva vstup v nasledovnom formáte:

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

Ak tvoj program dostane verdikt **Accepted**, sample grader vypíše text Accepted: q , kde q je počet volaní funkcie ask, ktoré tvoj program spravil.

Ak tvoj program dostane verdikt **Wrong Answer**, sample grader vypíše text Wrong

Answer: MSG, kde MSG je jedna z nasledujúcich správ:

- `answered not exactly once`: Funkciu `answer` tvoj program buď vôbec nezavolať, alebo ju zavolať aspoň dvakrát.
- `w is invalid`: Buď má tvoje pole `w` zadané do funkcie `ask` dĺžku inú ako M , alebo pre niektoré i platí, že `w[i]` nie je ani 0 ani 1.
- `more than 100 calls to ask`: Tvoj program sa pokúsil funkciu `ask` zavolať viac ako 100-krát.
- `{s, t} is wrong`: Funkciu `answer` tvoj program zavolať s nesprávnou dvojicou `s` a `t`.