



Jalan Tol

Di Jepang, kota-kota dihubungkan dengan sebuah jaringan jalan tol. Jaringan ini terdiri dari N kota dan M jalan tol. Setiap jalan tol menghubungkan sepasang kota berbeda. Tidak ada dua jalan tol yang menghubungkan pasangan kota yang sama. Kota-kota tersebut dinomori dari 0 hingga $N - 1$, dan jalan tol dinomori dari 0 hingga $M - 1$. Anda dapat menggunakan jalan tol tersebut dalam dua arah. Anda dapat pergi dari kota manapun ke kota manapun menggunakan jalan tol tersebut.

Sebuah biaya akan dikenakan untuk menggunakan setiap jalan tol. Biaya sebuah jalan tol tergantung pada kondisi **lalu lintas** pada jalan tersebut. Kondisi lalu lintas tersebut adalah **lancar** atau **macet**. Ketika lalu lintas sedang lancar, biaya jalan tol tersebut adalah A yen (mata uang Jepang). Ketika lalu lintas sedang macet, biaya untuk jalan tol tersebut adalah B yen. Dijamin bahwa $A < B$. Perhatikan bahwa Anda mengetahui nilai A dan B .

Anda memiliki sebuah mesin yang apabila diberikan kondisi lalu lintas di semua jalan tol, akan menghitung biaya termurah yang harus dibayar untuk berpergian antara kota S dan kota T ($S \neq T$), dalam kondisi lalu lintas yang diberikan.

Namun, mesin tersebut hanyalah sebuah *prototype*. Nilai dari S dan T telah ditetapkan (*hardcoded* di dalam mesin tersebut) dan Anda tidak mengetahui nilai tersebut. Anda ingin menentukan nilai S dan T . Untuk melakukannya, Anda berencana untuk memberikan beberapa kondisi lalu lintas kepada mesin tersebut dan menggunakan biaya yang dikeluarkan untuk menyimpulkan S dan T . Karena memberikan kondisi lalu lintas mahal, Anda tidak ingin menggunakan mesin tersebut terlalu sering.

Detail implementasi

Anda harus mengimplementasikan prosedur berikut:

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

- N : Jumlah kota.
- U dan V : array dengan panjang M , dengan M adalah banyaknya jalan tol. Untuk setiap i ($0 \leq i \leq M - 1$), jalan tol ke- i menghubungkan kota $U[i]$ dan $V[i]$.
- A : Biaya tol apabila kondisi lalu lintas sedang lancar.
- B : Biaya tol apabila kondisi lalu lintas sedang macet.
- Prosedur ini akan dipanggil tepat sekali untuk setiap kasus uji.

- Perhatikan bahwa nilai M adalah panjang array dan bisa didapatkan seperti yang dicantumkan pada pemberitahuan implementasi.

Prosedur `find_pair` dapat memanggil fungsi berikut:

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

- Panjang w harus tepat M . Array w menjelaskan kondisi lalu lintas.
- Untuk setiap i ($0 \leq i \leq M - 1$), $w[i]$ menjelaskan kondisi lalu lintas di jalan tol ke- i . Nilai dari $w[i]$ haruslah 0 atau 1.
 - $w[i] = 0$ berarti bahwa kondisi lalu lintas jalan tol ke- i sedang lancar.
 - $w[i] = 1$ berarti bahwa kondisi lalu lintas jalan tol ke- i sedang macet.
- Fungsi ini mengembalikan biaya termurah untuk berpergian antara kota S dan T , dengan kondisi lalu lintas yang diberikan oleh w .
- Fungsi ini dapat dipanggil paling banyak 100 kali (untuk setiap kasus uji).

`find_pair` harus memanggil prosedur berikut untuk memberikan jawaban:

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

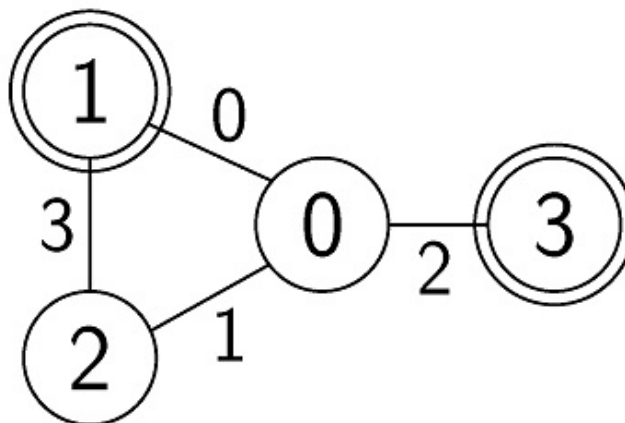
- s dan t adalah pasangan kota S dan T (urutan tidak berpengaruh).
- prosedur ini harus dipanggil tepat sekali.

Apabila beberapa kondisi di atas tidak terpenuhi, program Anda akan dinilai sebagai **Wrong Answer**. Selain itu, program Anda akan dinilai sebagai **Accepted** dan nilai Anda akan dihitung berdasarkan jumlah pemanggilan `ask` (lihat subsoal).

Contoh

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

Grader memanggil `find_pair(4, [0, 0, 0, 1], [1, 2, 3, 2], 1, 3)`.



Pada gambar di atas, *edge* dengan nomor i merupakan jalan tol ke- i .

Beberapa pemanggilan `ask` yang mungkin dan nilai kembaliannya:

Call	Return
<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

Untuk pemanggilan fungsi `ask([0, 0, 0, 0])`, kondisi lalu lintas untuk semua jalan tol adalah lancar dan biaya yang diperlukan adalah 1. Rute termurah dari $S = 1$ ke $T = 3$ adalah $1 \rightarrow 0 \rightarrow 3$. Biaya yang diperlukan untuk rute ini adalah 2. jadi, fungsi ini mengembalikan 2.

Untuk jawaban yang benar, prosedur `find_pair` harus memanggil `answer(1, 3)` atau `answer(3, 1)`.

Berkas `sample-01-in.txt` di dalam zip lampiran merupakan masukan dari contoh ini. Contoh masukan lain juga tersedia di dalam paket tersebut.

Batasan

- $2 \leq N \leq 90\,000$
- $1 \leq M \leq 130\,000$
- $1 \leq A < B \leq 1\,000\,000\,000$
- Untuk setiap $0 \leq i \leq M - 1$
 - $0 \leq U[i] \leq N - 1$
 - $0 \leq V[i] \leq N - 1$
 - $U[i] \neq V[i]$
- $(U[i], V[i]) \neq (U[j], V[j])$ dan $(U[i], V[i]) \neq (V[j], U[j])$ ($0 \leq i < j \leq M - 1$)
- Anda dapat pergi dari kota manapun ke kota manapun menggunakan jalan tol tersebut.
- $0 \leq S \leq N - 1$
- $0 \leq T \leq N - 1$
- $S \neq T$

Pada soal ini, grader TIDAK adaptif. Ini berarti bahwa S dan T telah ditetapkan pada awal berjalannya grader dan tidak tergantung pada pertanyaan yang ditanyakan oleh solusi Anda.

Subsoal

1. (5 poin) salah satu dari S atau T adalah 0, $N \leq 100$, $M = N - 1$
2. (7 poin) salah satu dari S atau T adalah 0, $M = N - 1$
3. (6 poin) $M = N - 1$, $U[i] = i$, $V[i] = i + 1$ ($0 \leq i \leq M - 1$)
4. (33 poin) $M = N - 1$
5. (18 poin) $A = 1$, $B = 2$
6. (31 poin) Tidak ada batasan khusus

Asumsikan program Anda dinilai sebagai **Accepted** dan melakukan X pemanggilan ask. Maka nilai Anda P untuk kasus uji, tergantung pada nomor subtask, akan dihitung sebagai berikut:

- Subsoal 1. $P = 5$.
- Subsoal 2. Jika $X \leq 60$, $P = 7$. Selain itu $P = 0$.
- Subsoal 3. Jika $X \leq 60$, $P = 6$. Selain itu $P = 0$.
- Subsoal 4. Jika $X \leq 60$, $P = 33$. Selain itu $P = 0$.
- Subsoal 5. Jika $X \leq 52$, $P = 18$. Selain itu $P = 0$.
- Subsoal 6.
 - Jika $X \leq 50$, $P = 31$.
 - Jika $51 \leq X \leq 52$, $P = 21$.
 - Jika $53 \leq X$, $P = 0$.

Perhatikan bahwa nilai Anda untuk setiap subsoal adalah nilai minimum dari semua kasus uji yang ada di dalam subsoal tersebut.

Contoh grader

Contoh grader membaca input dengan format sebagai berikut:

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

Apabila program Anda dinilai sebagai **Accepted**, contoh grader akan mencetak Accepted: q, dengan q adalah jumlah pemanggilan ask.

Apabila program Anda dinilai sebagai **Wrong Answer**, contoh grader akan mencetak Wrong Answer: MSG, dengan MSG adalah salah satu dari:

- answered not exactly once: Prosedur answer tidak dipanggil tepat sekali.
- w is invalid: Panjang dari w yang diberikan kepada ask bukan M atau $w[i]$ bukan 0 atau 1 untuk sebuah i ($0 \leq i \leq M - 1$).
- more than 100 calls to ask: Fungsi ask dipanggil lebih dari 100 kali.
- {s, t} is wrong: Prosedur answer dipanggil dengan pasangan s dan t yang salah.