



Vé đường cao tốc

Ở Nhật Bản, các thành phố được kết nối bằng mạng lưới các con đường cao tốc. Mạng lưới này gồm N thành phố và M con đường cao tốc. Mỗi con đường cao tốc nối một cặp 2 thành phố khác nhau. Không có 2 con đường cao tốc nối cùng một cặp hai thành phố. Các thành phố được đánh số từ 0 đến $N - 1$, và các con đường cao tốc được đánh số từ 0 đến $M - 1$. Bạn có thể lái xe trên mỗi con đường cao tốc theo cả 2 chiều. Bạn có thể đi từ một thành phố bất kỳ sang một thành phố bất kỳ khác qua những con đường cao tốc.

Phải mua vé khi lái xe trên mỗi con đường cao tốc. Giá vé cho mỗi con đường cao tốc phụ thuộc vào **độ tắc nghẽn** của con đường. Độ tắc nghẽn có thể là **thưa** hoặc **dày**. Khi độ tắc nghẽn là thưa, giá vé là A yen (đơn vị tiền tệ Nhật Bản). Khi độ tắc nghẽn là dày, giá vé là B yen. Đảm bảo rằng $A < B$. Chú ý là bạn biết giá trị của A và B .

Bạn có một cỗ máy, khi được cho độ tắc nghẽn của tất cả các con đường cao tốc, nó tính ra tổng giá vé nhỏ nhất phải trả khi di chuyển giữa cặp 2 thành phố S và T ($S \neq T$), trong trạng thái giao thông đó.

Tuy nhiên cỗ máy này mới chỉ ở dạng thử nghiệm. Giá trị S và T cố định (tức là được cài đặt cố định vào cỗ máy) và bạn không biết giá trị này. Bạn phải xác định giá trị của S và T . Để thực hiện điều này, bạn lên kế hoạch nhập vào cỗ máy một số bộ giá trị các độ tắc nghẽn, và sử dụng tổng giá vé mà cỗ máy tính ra để xác định S và T . Vì việc nhập vào các độ tắc nghẽn là tốn kém, bạn không muốn sử dụng cỗ máy nhiều lần.

Chi tiết cài đặt

Bạn phải cài đặt thủ tục sau:

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

- N : số lượng các thành phố.
- U và V : các mảng có độ dài M , với M là số lượng các con đường cao tốc nối giữa các thành phố. Đối với mỗi i ($0 \leq i \leq M - 1$), con đường cao tốc thứ i nối 2 thành phố $U[i]$ và $V[i]$.
- A : giá vé trên con đường cao tốc với độ tắc nghẽn là thưa.
- B : giá vé trên con đường cao tốc với độ tắc nghẽn là dày.
- Thủ tục này được gọi đúng một lần cho mỗi test case.
- Chú ý rằng giá trị M là độ dài của mảng, và cách lấy độ dài mảng xem trong Lưu ý

cài đặt.

Thủ tục `find_pair` có thể gọi hàm sau:

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

- Độ dài của w phải đúng bằng M . Mảng w mô tả độ tắc nghẽn.
- Với mỗi i ($0 \leq i \leq M - 1$), $w[i]$ mô tả độ tắc nghẽn trên con đường cao tốc thứ i . Giá trị của $w[i]$ hoặc là 0 hoặc là 1.
 - $w[i] = 0$ nghĩa là độ tắc nghẽn trên con đường cao tốc thứ i là thưa.
 - $w[i] = 1$ nghĩa là độ tắc nghẽn trên con đường cao tốc thứ i là dày.
- Hàm này trả lại giá trị là tổng giá vé nhỏ nhất khi di chuyển giữa 2 thành phố S và T , với điều kiện trạng thái giao thông được xác định bởi w .
- Hàm này được gọi tối đa 100 lần (cho mỗi test case).

`find_pair` phải gọi thủ tục sau để báo cáo kết quả:

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

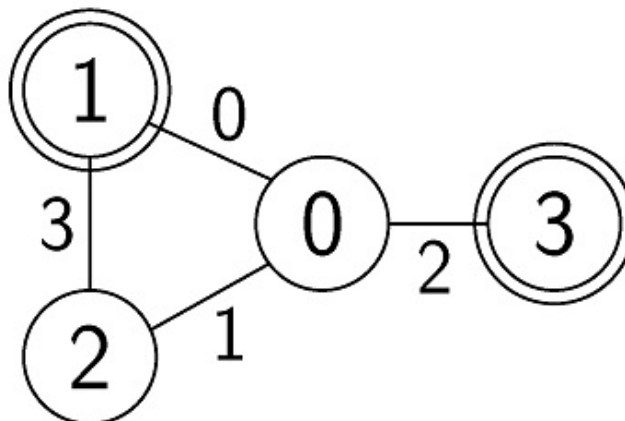
- s và t là 2 cặp S và T (thứ tự không quan trọng).
- Thủ tục này chỉ được gọi đúng một lần.

Nếu điều kiện nào đó trong những điều kiện trên không thỏa mãn, chương trình của bạn bị đánh giá **Wrong Answer**. Ngược lại, chương trình của bạn được đánh giá **Accepted** và điểm số của bạn được tính toán dựa trên số lượng lần gọi `ask` (xem phần Subtasks).

Ví dụ

Giả sử $N = 4$, $M = 4$, $U = [0, 0, 0, 1]$, $V = [1, 2, 3, 2]$, $A = 1$, $B = 3$, $S = 1$, và $T = 3$.

Trình chấm gọi `find_pair(4, [0, 0, 0, 1], [1, 2, 3, 2], 1, 3)`.



Trong hình trên, các cạnh với số thứ tự i tương ứng với con đường cao tốc thứ i . Một số lời gọi ask có thể và giá trị trả về tương ứng được liệt kê dưới đây:

Lời gọi	Giá trị trả về
ask([0, 0, 0, 0])	2
ask([0, 1, 1, 0])	4
ask([1, 0, 1, 0])	5
ask([1, 1, 1, 1])	6

Với lời gọi ask([0, 0, 0, 0]), độ tắc nghẽn của tất cả các con đường cao tốc là thưa, và giá vé cho mỗi con đường là 1. Tuyến đường chi phí thấp nhất từ $S = 1$ tới $T = 3$ là $1 \rightarrow 0 \rightarrow 3$. Tổng giá vé trên tuyến đường này là 2. Do đó, hàm này trả về giá trị 2.

Để đưa ra câu trả lời đúng, thủ tục find_pair phải gọi answer(1, 3) hoặc answer(3, 1).

File sample-01-in.txt trong gói file zip đính kèm tương ứng với ví dụ này. Gói file zip này cũng chứa một số ví dụ khác.

Hạn chế

- $2 \leq N \leq 90\,000$
- $1 \leq M \leq 130\,000$
- $1 \leq A < B \leq 1\,000\,000\,000$
- Với mỗi $0 \leq i \leq M - 1$
 - $0 \leq U[i] \leq N - 1$
 - $0 \leq V[i] \leq N - 1$
 - $U[i] \neq V[i]$
- $(U[i], V[i]) \neq (U[j], V[j])$ và $(U[i], V[i]) \neq (V[j], U[j])$ ($0 \leq i < j \leq M - 1$)
- Bạn có thể đi từ một thành phố bất kỳ đến bất kỳ thành phố nào còn lại nhờ sử dụng các con đường cao tốc.
- $0 \leq S \leq N - 1$
- $0 \leq T \leq N - 1$
- $S \neq T$

Trong bài này, trình chấm không có tính thích nghi. Điều này có nghĩa là S và T được cố định ngay từ đầu khi trình chấm chạy và chúng không phụ thuộc vào các truy vấn bởi chương trình của bạn.

Subtasks

1. (5 điểm) một trong S hoặc T là 0, $N \leq 100$, $M = N - 1$

2. (7 điểm) một trong S hoặc T is 0, $M = N - 1$
3. (6 điểm) $M = N - 1$, $U[i] = i$, $V[i] = i + 1$ ($0 \leq i \leq M - 1$)
4. (33 điểm) $M = N - 1$
5. (18 điểm) $A = 1$, $B = 2$
6. (31 điểm) Không có hạn chế nào khác

Giả sử chương trình của bạn được đánh giá là **Accepted**, và gọi ask X lần. Phụ thuộc vào số thứ tự subtask, điểm số P của bạn cho test case này được tính toán như sau:

- Subtask 1. $P = 5$.
- Subtask 2. Nếu $X \leq 60$, $P = 7$. Ngược lại $P = 0$.
- Subtask 3. Nếu $X \leq 60$, $P = 6$. Ngược lại $P = 0$.
- Subtask 4. Nếu $X \leq 60$, $P = 33$. Ngược lại $P = 0$.
- Subtask 5. Nếu $X \leq 52$, $P = 18$. Ngược lại $P = 0$.
- Subtask 6.
 - Nếu $X \leq 50$, $P = 31$.
 - Nếu $51 \leq X \leq 52$, $P = 21$.
 - Nếu $53 \leq X$, $P = 0$.

Chú ý là điểm số của bạn cho mỗi subtask là điểm số nhỏ nhất của các test case trong subtask đó.

Trình chấm mẫu

Trình chấm mẫu đọc dữ liệu đầu vào theo khuôn dạng sau:

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

Nếu chương trình của bạn được đánh giá là **Accepted**, thì trình chấm mẫu sẽ in ra Accepted: q , với q là số lượng lần gọi ask.

Nếu chương trình của bạn được đánh giá là **Wrong Answer**, trình chấm mẫu in ra Wrong Answer: MSG, trong đó MSG có thể là:

- answered not exactly once: Thủ tục answer không phải chỉ được gọi đúng 1 lần.
- w is invalid: Độ dài của w trong ask khác M hoặc $w[i]$ khác với 0 và 1 với i nào đó ($0 \leq i \leq M - 1$).
- more than 100 calls to ask: Hàm ask bị gọi nhiều hơn 100 lần.
- {s, t} is wrong: Thủ tục answer được gọi với cặp s and t sai.