B. Swapping Cities

Time limit	2 s
Memory limit	512 MB

Mô tả bài toán

Có N thành phố ở Indonesia, được đánh số từ 0 đến $N-1$. Ngoài ra, còn có M con đường hai chiều, được đánh số từ 0 đến $M-1$. Mỗi con đường nối hai thành phố khác nhau. Con đường thứ i kết nối thành phố thứ $U[i]$với thành phố thứ $V[i]$ và tiêu thụ $W[i]$ đơn vị xăng khi đi qua bằng ô tô. Các thành phố được kết nối với nhau bảo đảm có thể đi lại giữa bất kỳ cặp thành phố nào thông qua các con đường này.

Đối với mỗi ngày trong Q ngày tiếp theo, một cặp thành phố muốn thiết lập mối quan hệ chính trị. Cụ thể, vào ngày thứ j, thành phố thứ $X[j]$ muốn thiết lập mối quan hệ chính trị với thành phố thứ $Y[j]$. Để thực hiện điều này, thành phố thứ $X[j]$ sẽ cử một người đại diện đi đến thành phố thứ $Y[j]$bằng ô tô. Tương tự, thành phố thứ $Y[j]$ cũng sẽ cử một người đại diện đi đến thành phố thứ $X[j]$ bằng ô tô.

Để tránh ùn tắc, cả hai ô tô không nên gặp nhau vào bất kì thời điểm nào. Cụ thể, cả hai ô tô không nên ở trong cùng một thành phố vào cùng một thời điểm. Cả hai ô tô cũng không nên đi cùng một con đường theo hướng ngược nhau cùng một thời điểm. Hơn nữa, khi ô tô đi trên một con đường thì phải đi hết con đường và đi đến thành phố đích (nói cách khác, ô tô không được phép quay đầu xe ở giữa đường). Tuy nhiên, ô tô được phép đến cùng một thành phố hay đi một con đường nhiều hơn một lần. Ngoài ra, ô tô cũng có thể chờ ở bất kì thành phố nào vào bất kỳ thời điểm nào.

Vì ô tô có dung tích bình nhiên liệu lớn sẽ đắt tiền nên cả hai thành phố muốn chọn tuyến đường cho cả hai ô tô sao cho dung tích bình nhiên liệu lớn nhất của hai ô tô là nhỏ nhất. Ở mỗi thành phố đều có các trạm xăng với nguồn cung cấp vô hạn xăng, do đó dung tích bình nhiên liệu mà ô tô cần là mức tiêu thụ xăng **lớn nhất** trong tất cả các con đường mà ô tô đi qua.

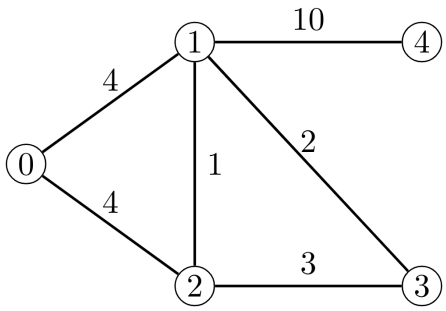
Yêu cầu

Bạn phải cài đặt hàm `init` và hàm `getMinimumFuelCapacity`.

- `init(N, M, U, V, W)` - Hàm này sẽ được gọi bởi trình chấm chính xác một lần trước bất kì lời gọi hàm `getMinimumFuelCapacity`.
  - N: Một số nguyên biểu diễn số lượng thành phố.
  - M: Một số nguyên biểu diễn số lượng con đường.
  - U: Một mảng gồm M số nguyên biểu diễn đầu mút thứ nhất của các con đường.
  - V: Một mảng gồm M số nguyên biểu diễn đầu mút thứ hai của các con đường.
  - W: Một mảng gồm M số nguyên biểu diễn mức tiêu thụ xăng của các con đường.
- `getMinimumFuelCapacity(X, Y)` - Hàm này sẽ được gọi bởi trình chấm chính xác Q lần.
  - X: Một số nguyên biểu diễn thành phố thứ nhất.
  - Y: Một số nguyên biểu diễn thành phố thứ hai.
  - Hàm này phải trả về một số nguyên biểu diễn giá trị nhỏ nhất của dung tích bình nhiên liệu lớn nhất của cả hai ô tô mà một cho người đại diện từ thành phố thứ X có thể đến thành phố thứ Y và một cho người đại diện từ thành phố thứ Y có thể đi đến thành phố thứ X theo các quy tắc được giải thích trong đề bài hoặc -1 nếu không tồn tại cách thực hiện.

Ví dụ

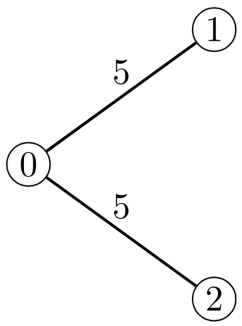
Trong ví dụ thứ nhất, $N=5$, $M=6$, $U=[0,0,1,1,1,2]$, $V=[1,2,2,3,4,3]$, $W=[4,4,1,2,10,3]$, $Q=3$, $X=[1,2,0]$, $Y=[2,4,1]$. Ví dụ được minh họa bằng hình dưới đây:



Trình chấm sẽ gọi khởi tạo `init(5, 6, [0, 0, 1, 1, 1, 2], [1, 2, 2, 3, 4, 3], [4, 4, 1, 2, 10, 3])`. Sau đó, trình chấm sẽ gọi như sau:

- `getMinimumFuelCapacity(1, 2)`. Đầu tiên, ô tô từ thành phố thứ nhất có thể đến thành phố thứ ba. Tiếp theo, ô tô từ thành phố thứ hai có thể đi đến thành phố thứ nhất và ô tô từ thành phố thứ ba có thể đến thành phố thứ hai. Do đó, bình nhiên liệu tối đa của hai ô tô là 3 đơn vị nhiên liệu (cần để đi từ thành phố thứ ba đến thành phố thứ hai). Không có tuyến đường nào để dung tích bình nhiên liệu có thể nhỏ hơn, do đó hàm phải trả về 3.
- `getMinimumFuelCapacity(2, 4)`. Bất kỳ ô tô nào đi đến hoặc đi từ thành phố thứ tư đều phải cần 10 đơn vị dung tích nhiên liệu, do đó hàm sẽ trả về 10.
- `getMinimumFuelCapacity(0, 1)`. Hàm sẽ trả về 4.

Trong ví dụ thứ hai, $N = 3$, $M = 2$, $U = [0, 0]$, $V = [1, 2]$, $W = [5, 5]$, $Q = 1$, $X = [1]$, $Y = [2]$. Ví dụ được minh họa bằng hình dưới đây:



Trình chấm sẽ gọi khởi tạo `init(3, 2, [0, 0], [1, 2], [5, 5])`. Sau đó, trình chấm sẽ gọi như sau:

- `getMinimumFuelCapacity(1, 2)`. Không thể xảy ra trường hợp ô tô ở thành phố thứ nhất đi đến thành phố thứ hai mà không gặp ô tô kia tại một thời điểm nào đó, do đó hàm sẽ trả về -1.

Các ràng buộc

- $2 \le N \le 100\,000$.
- $N - 1 \le M \le 200\,000$.
- $0 \le U[i] < V[i] < N$.
- Có nhiều nhất một con đường giữa mỗi cặp thành phố.
- Có thể đi lại giữa bất kỳ cặp thành phố nào thông qua các con đường.
- $1 \le W[i] \le 10^9$.
- $1 \le Q \le 200\,000$.
- $0 \le X[j] < Y[j] < N$.

Subtask 1 (6 điểm)

- Mỗi thành phố là điểm đầu mút của nhiều nhất hai con đường.

Subtask 2 (7 điểm)

- $M = N - 1$.
- $U[i] = 0$.

Subtask 3 (17 điểm)

- $Q \leq 5$.
- $N \leq 1\,000$.
- $M \leq 2\,000$.

Subtask 4 (20 điểm)

- $Q \leq 5$.

Subtask 5 (23 điểm)

- $M = N - 1$.

Subtask 6 (27 điểm)

- Không có điều kiện ràng buộc gì thêm.

Trình chấm mẫu

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

```
N M
U[0] V[0] W[0]
U[1] V[1] W[1]
.
.
.
U[M-1] V[M-1] W[M-1]
Q
X[0] Y[0]
X[1] Y[1]
.
.
.
X[Q-1] Y[Q-1]
```

Với mỗi lời gọi `getMinimumFuelCapacity`, trình chấm mẫu in ra giá trị trả về bởi hàm.