

C. Fun Tour

Time limit	2 s
Memory limit	512 MB

Mô tả bài toán

Có N điểm tham quan trong công viên chủ đề lớn nhất ở Jakarta, được đánh số từ 0 đến $N - 1$. Các điểm tham quan này được nối với nhau bằng $N - 1$ con đường hai chiều sao cho giữa hai điểm tham quan bất kì có duy nhất một cách đi thông qua các con đường này. Các con đường được đánh số từ 0 đến $N - 2$. Con đường thứ i nối điểm tham quan thứ $A[i]$ và điểm tham quan thứ $B[i]$ và mất 1 giờ để đi bộ. Để tránh tắc đường, mỗi điểm tham quan là điểm đầu mút của nhiều nhất 3 con đường.

Bạn muốn tạo một chuyến đi thăm qua tất cả các điểm tham quan, mỗi điểm đúng một lần. Việc đi qua nhiều con đường để di chuyển từ điểm tham quan này đến một điểm tham quan khác thì buồn chán. Vì vậy, bạn muốn tạo một chuyến đi tham quan vui vẻ bằng cách xếp thứ tự các điểm tham quan sao cho thời gian cần để di chuyển đến điểm tham quan tiếp theo thì không lớn hơn thời gian cần để di chuyển đến điểm tham quan trước đó. Nói một cách khác, bạn muốn tìm một dãy $P[0], P[1], \dots, P[N - 1]$ chứa tất cả các số nguyên từ 0 to $N - 1$, mỗi số đúng một lần sao cho thời gian cần để di chuyển từ điểm tham quan thứ $P[i]$ tới điểm tham quan thứ $P[i + 1]$ không lớn hơn thời gian cần để di chuyển từ điểm tham quan thứ $P[i - 1]$ tới điểm tham quan thứ $P[i]$, với $0 < i < N - 1$.

Bạn không có bản đồ đầy đủ các điểm tham quan. Vì vậy, bạn phải hỏi trung tâm thông tin một số câu hỏi để có thể tạo được chuyến tham quan vui vẻ. Bạn có thể hỏi tối đa Q câu hỏi, mỗi câu hỏi có hai tham số X và Y , trong đó $0 \leq X, Y < N$. Mỗi câu hỏi thuộc một trong hai dạng sau:

- Cần bao nhiêu giờ để đi từ điểm tham quan thứ X đến điểm tham quan thứ Y ? Đặc biệt, nếu $X = Y$, thì câu trả lời là 0.
- Có bao nhiêu điểm tham quan Z mà để đi từ điểm tham quan thứ X đến điểm tham quan thứ Z bạn phải đi qua điểm tham quan thứ Y ? Điểm tham quan thứ Y cũng được tính. Đặc biệt, nếu $X = Y$, câu trả lời là N .

Yêu cầu

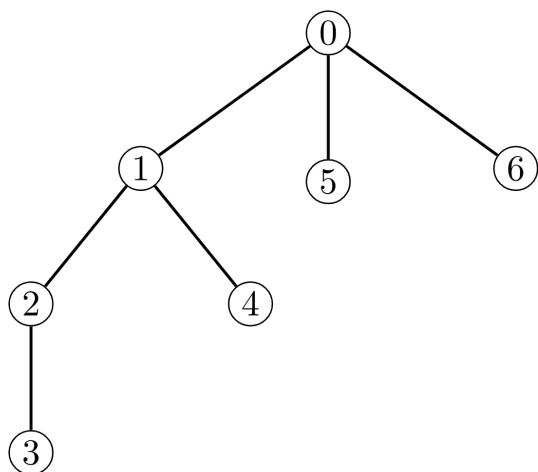
Bạn phải cài đặt hàm `createFunTour`:

- `createFunTour(N, Q)` - Hàm này sẽ được gọi bởi trình chấm đúng một lần.
 - N : Một số nguyên biểu diễn số điểm tham quan.
 - Q : Một số nguyên biểu diễn số lượng tối đa các câu hỏi.
 - Hàm này được phép gọi hai hàm của trình chấm:
 - `hoursRequired(X, Y)`
 - X : Một số nguyên biểu diễn điểm tham quan thứ nhất.
 - Y : Một số nguyên biểu diễn điểm tham quan thứ hai.
 - Hàm này trả về một số nguyên biểu diễn số giờ cần để đi từ điểm tham quan thứ X đến điểm tham quan thứ Y .
 - Nếu một trong hai X hoặc Y không phải là số nguyên có giá trị từ 0 đến $N - 1$, thì bạn sẽ nhận được kết quả là WA.
 - `attractionsBehind(X, Y)`
 - X : Một số nguyên biểu diễn điểm tham quan thứ nhất.
 - Y : Một số nguyên biểu diễn điểm tham quan thứ hai.

- Hàm này trả về một số nguyên biểu diễn số lượng điểm tham quan Z mà để đi từ điểm tham quan thứ X đến điểm tham quan thứ Z bạn phải đi qua điểm tham quan thứ Y .
- Nếu một trong hai X hoặc Y không phải là số nguyên có giá trị từ 0 đến $N - 1$, thì bạn sẽ nhận kết quả là WA.
- Hàm này bắt buộc phải trả về một mảng chứa N số nguyên biểu diễn một hoán vị của các điểm tham quan trong chuyến đi vui vẻ.

Ví dụ

Trong ví dụ sau đây, $N = 7$, $Q = 400\,000$, $A = [0, 0, 0, 1, 1, 2]$, và $B = [1, 5, 6, 2, 4, 3]$. Ví dụ này được minh họa bởi hình dưới đây:



Chương trình chấm sẽ gọi hàm `createFunTour(7, 400000)`.

- Nếu thí sinh gọi hàm `hoursRequired(3, 5)`, thì hàm này sẽ trả lại giá trị 4.
- Nếu thí sinh gọi hàm `hoursRequired(5, 4)`, thì hàm này sẽ trả lại giá trị 3.
- Nếu thí sinh gọi hàm `attractionsBehind(5, 1)`, thì hàm này sẽ trả lại giá trị 4. Để đi từ điểm tham quan thứ năm đến các điểm tham quan thứ nhất, thứ hai, thứ ba, thứ bốn, thì bạn phải đi qua điểm tham quan thứ nhất.
- Nếu thí sinh gọi hàm `attractionsBehind(1, 5)`, thì hàm này sẽ trả lại giá trị 1.
- Thí sinh có thể trả lại kết quả `[3, 6, 4, 5, 2, 0, 1]` bởi vì số giờ cần thiết để đi đến điểm tham quan tiếp theo dựa vào thứ tự trên là `[4, 3, 3, 3, 2, 1]`.

Các ràng buộc

- $2 \leq N \leq 100\,000$.
- $Q = 400\,000$.
- Có thể di chuyển giữa bất kỳ một cặp điểm tham quan nào thông qua các con đường trên.
- Mỗi điểm tham quan là điểm đầu mút của không quá 3 con đường.

Subtask 1 (10 điểm)

- $N \leq 17$.

Subtask 2 (16 điểm)

- $N \leq 500$.

Subtask 3 (21 điểm)

- Có một con đường nối điểm tham quan thứ i và điểm tham quan thứ $\lfloor \frac{i-1}{2} \rfloor$, với mọi $1 \leq i < N$.

Subtask 4 (19 điểm)

- Có ít nhất một điểm tham quan T sao cho với mọi $0 \leq i < N$, `hoursRequired(T, i) < 30` và tồn tại một đoạn $[L[i], R[i]]$ ($0 \leq L[i] \leq i \leq R[i] < N$) thỏa mãn các ràng buộc sau:
 - Bạn phải qua đi qua điểm tham quan thứ i để đi từ điểm tham quan thứ T đến điểm tham quan thứ j khi và chỉ khi $L[i] \leq j \leq R[i]$.
 - Nếu $L[i] < i$, thì có đúng một điểm tham quan X sao cho:
 - $L[i] \leq X < i$.
 - Có một con đường nối điểm tham quan thứ i và điểm tham quan thứ X .
 - Nếu $i < R[i]$, thì có đúng một điểm tham quan Y sao cho:
 - $i < Y \leq R[i]$.
 - Có một con đường nối điểm tham quan thứ i và điểm tham quan thứ Y .

Subtask 5 (34 điểm)

- Không có ràng buộc gì thêm.

Trình chấm mẫu

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

```
N Q
A[0] B[0]
A[1] B[1]
.
.
.
A[N-2] B[N-2]
```

Trình chấm mẫu ghi các số nguyên trả về từ hàm `createFunTour` nếu nó trả về đúng một mảng N số nguyên biểu diễn một hoán vị của các điểm tham quan trong chuyến đi vui vẻ và tổng số lần gọi cả hai hàm `hoursRequired` và `attractionsBehind` không quá Q . Ngược lại, nó sẽ in ra một thông báo là câu trả lời sai.