



Stations (stations)

Mạng trực Internet của Singapore (SIB) bao gồm n trạm, được đánh **chỉ số** từ 0 đến $n - 1$. Có $n - 1$ đường truyền dữ liệu hai chiều đánh số từ 0 đến $n - 2$. Mỗi đường truyền kết nối hai trạm phân biệt. Hai trạm nối với nhau bởi một đường truyền được gọi là trạm kề nhau.

Một đường đi từ trạm x đến trạm y là một dãy các trạm phân biệt a_0, a_1, \dots, a_p , thỏa mãn $a_0 = x$, $a_p = y$, và mỗi cặp hai trạm liên tiếp trên đường đi là kề nhau. Có **đúng một** đường đi từ trạm x bất kỳ đến trạm y bất kỳ khác.

Mỗi trạm x bất kỳ có thể tạo ra một gói tin (chứa dữ liệu) và gửi nó tới trạm y bất kỳ khác, được gọi là trạm **đích**. Gói tin này phải được định tuyến trên đường đi duy nhất từ x đến y như sau. Xét trạm z là trạm đang giữ một gói tin, trạm y là trạm đích ($z \neq y$). Trong trường hợp này, trạm z sẽ thực hiện:

1. thực thi **hàm định tuyến** để xác định trạm kề của trạm z nằm trên đường đi duy nhất từ z đến y , và
2. chuyển tiếp gói tin cho trạm kề này.

Tuy nhiên, các trạm có bộ nhớ giới hạn và không thể lưu trữ toàn bộ danh sách các đường truyền của SIB để sử dụng trong hàm định tuyến.

Nhiệm vụ của bạn là lên kế hoạch định tuyến cho SIB, bao gồm 2 hàm.

- Hàm thứ nhất nhận vào giá trị n , danh sách các đường truyền dữ liệu của SIB và một số nguyên $k \geq n - 1$. Hàm sẽ gán **nhãn** mỗi trạm là một số nguyên **duy nhất** trong khoảng từ 0 đến k , bao gồm cả hai đầu mút.
- Hàm thứ hai là hàm định tuyến, dùng để triển khai tại tất cả các trạm sau khi đã được gán nhãn. Hàm này **chỉ** nhận các dữ liệu đầu vào sau:
 - s , **nhãn** của trạm đang giữ gói tin,
 - t , **nhãn** của trạm đích ($t \neq s$),
 - c , danh sách **nhãn** của tất cả các trạm kề với trạm có nhãn s .

Hàm trả về **nhãn** của trạm kề với trạm có nhãn s mà gói tin cần được chuyển tiếp tới.

Trong một subtask, điểm của bài làm của bạn phụ thuộc vào giá trị lớn nhất của các nhãn được gán cho một trạm bất kỳ (nói chung, càng nhỏ càng tốt).

Chi tiết cài đặt

Bạn cần cài đặt các hàm sau:

```
int[] label(int n, int k, int[] u, int[] v)
```

- n : số trạm trong SIB.
- k : giá trị nhãn lớn nhất có thể sử dụng.
- u và v : các mảng kích thước $n - 1$ mô tả các đường truyền. Với mỗi i ($0 \leq i \leq n - 2$), đường truyền i nối hai trạm có chỉ số $u[i]$ và $v[i]$.
- Hàm cần trả về một mảng L với kích thước n . Với mỗi i ($0 \leq i \leq n - 1$), $L[i]$ là nhãn được gán cho trạm có chỉ số i . Tất cả các phần tử thuộc L phải đôi một khác nhau và nằm trong khoảng từ 0 đến k , bao gồm cả hai đầu mút.

```
int find_next_station(int s, int t, int[] c)
```

- s : nhãn của trạm đang giữ gói tin.
- t : nhãn của trạm đích của gói tin.
- c : một mảng chứa danh sách các nhãn của các trạm kề với trạm có nhãn s . Mảng c được sắp xếp theo thứ tự tăng dần.
- Hàm cần trả về nhãn của một trạm kề với trạm có nhãn s mà gói tin cần được chuyển tiếp tới.

Mỗi test chứa một hoặc nhiều kịch bản khác nhau (nghĩa là các mô tả cấu trúc SIB khác nhau). Với test chứa r kịch bản, một **chương trình** gọi các hàm trên được thực thi đúng hai lần như sau.

Trong lần chạy đầu tiên:

- hàm `label` được gọi r lần,
- các nhãn trả về được lưu trữ trong hệ thống chấm, và
- hàm `find_next_station` không được gọi.

Trong lần chạy thứ hai:

- hàm `find_next_station` có thể được gọi nhiều lần. Trong mỗi lời gọi, một kịch bản **bất kỳ** được lựa chọn, và trong kịch bản đó, nhãn được trả về bởi lời gọi hàm `label` được sử dụng làm dữ liệu vào cho hàm `find_next_station`.
- hàm `label` không được gọi.

Lưu ý rằng, mọi thông tin được lưu trữ trong các biến tĩnh (static) hay toàn cục (global) trong lần gọi đầu tiên sẽ không sử dụng được trong hàm `find_next_station`.

Ví dụ

Xét lời gọi hàm sau:

```
label(5, 10, [0, 1, 1, 2], [1, 2, 3, 4])
```

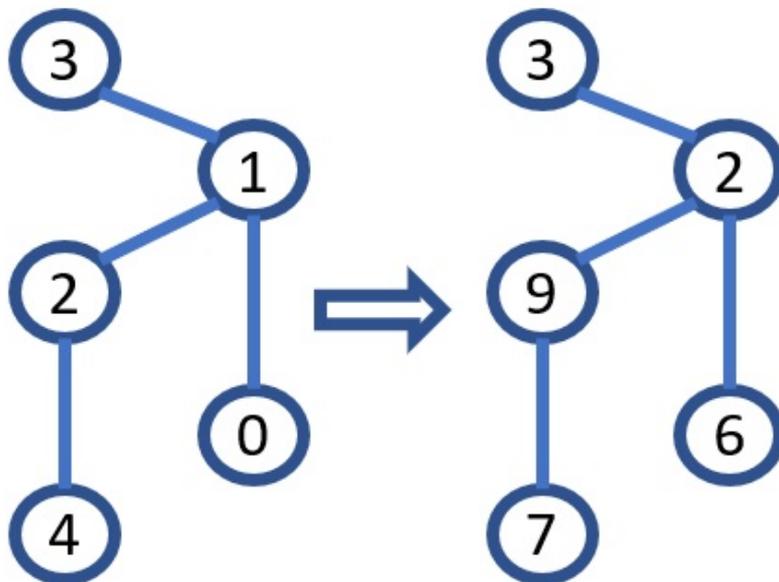
Có tổng cộng 5 trạm, và 4 đường truyền nối các cặp trạm có chỉ số (0, 1), (1, 2), (1, 3) và (2, 4).

Các nhãn có thể là các số nguyên trong khoảng từ 0 đến $k = 10$.

Để đưa ra cách gán nhãn sau:

Chỉ số	Nhãn
0	6
1	2
2	9
3	3
4	7

hàm `label` cần trả về `[6, 2, 9, 3, 7]`. Các số trong ảnh sau thể hiện các chỉ số (ảnh trái) và các giá trị nhãn được gán (ảnh phải).



Giả sử các nhãn đã gán như trên và xét lời gọi hàm sau:

```
find_next_station(9, 6, [2, 7])
```

Điều này có nghĩa là trạm đang giữ gói tin có nhãn 9, và trạm đích có nhãn 6. Nhãn của các trạm trên đường đến trạm đích là `[9, 2, 6]`. Vì vậy, hàm cần trả về giá trị 2, là nhãn của trạm mà gói tin cần được chuyển tiếp tới (trạm này có chỉ số là 1).

Xét một lời gọi hàm khác:

```
find_next_station(2, 3, [3, 6, 9])
```

Hàm cần trả về giá trị 3, vì trạm đích với nhãn 3 là trạm kề với trạm có nhãn 2, vậy nên có thể

chuyển trực tiếp gói tin tới trạm đích.

Ràng buộc

- $1 \leq r \leq 10$

Với mỗi lời gọi hàm `label`:

- $2 \leq n \leq 1000$
- $k \geq n - 1$
- $0 \leq u[i], v[i] \leq n - 1$ (với mọi $0 \leq i \leq n - 2$)

Với mỗi lời gọi hàm `find_next_station`, dữ liệu vào được chọn bất kỳ từ các lần gọi hàm `label` trước đó. Dựa trên các nhãn đã được gán thì:

- s và t là nhãn của hai trạm khác nhau.
- c là dãy các nhãn của tất cả các trạm kề với trạm có nhãn s , theo thứ tự tăng dần.

Với mỗi test, tổng độ dài của các mảng c được truyền vào hàm `find_next_station` không vượt quá 100 000 cho tất cả các kịch bản gộp lại.

Subtask

1. (5 điểm) $k = 1000$, không trạm nào có nhiều hơn 2 trạm kề.
2. (8 điểm) $k = 1000$, đường truyền i nối trạm $i + 1$ và $\lfloor \frac{i}{2} \rfloor$.
3. (16 điểm) $k = 1\,000\,000$, chỉ có nhiều nhất một trạm có nhiều hơn 2 trạm kề.
4. (10 điểm) $n \leq 8$, $k = 10^9$
5. (61 điểm) $k = 10^9$

Trong subtask 5, bạn có thể nhận được điểm thành phần. Gọi m là giá trị lớn nhất của các nhãn được trả về bởi hàm `label` trong tất cả các kịch bản. Điểm của bạn cho subtask này được tính dựa trên bảng sau:

Giá trị nhãn lớn nhất	Điểm
$m \geq 10^9$	0
$2000 \leq m < 10^9$	$50 \cdot \log_{5 \cdot 10^5}(\frac{10^9}{m})$
$1000 < m < 2000$	50
$m \leq 1000$	61

Trình chấm mẫu

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

- dòng 1: r

r nhóm dòng tiếp theo, mỗi nhóm mô tả một kịch bản. Định dạng của mỗi nhóm như sau:

- dòng 1: n k
- dòng $2 + i$ ($0 \leq i \leq n - 2$): $u[i]$ $v[i]$
- dòng $1 + n$: q : số lần gọi hàm `find_next_station`.
- dòng $2 + n + j$ ($0 \leq j \leq q - 1$): $z[j]$ $y[j]$ $w[j]$: **chỉ số** của các trạm trong lời gọi hàm `find_next_station` thứ j . Trạm $z[j]$ đang giữ gói tin, trạm $y[j]$ là trạm đích, và trạm $w[j]$ là trạm mà gói tin cần được chuyển tiếp đến.

Trình chấm mẫu in kết quả theo định dạng sau:

- dòng 1: m

r nhóm dòng tiếp theo mô tả các kịch bản tương ứng trong dữ liệu vào. Định dạng của mỗi nhóm dòng như sau:

- dòng $1 + j$ ($0 \leq j \leq q - 1$): **chỉ số** của trạm có **nhãn** được trả về bởi lời gọi hàm `find_next_station` thứ j trong kịch bản này.

Lưu ý rằng, trong mỗi lần chạy, trình chấm mẫu gọi cả hàm `label` và hàm `find_next_station`.