Simurgh

Theo truyền thuyết Ba Tư cổ đại ở Shahnameh, Zal, anh hùng huyền thoại người Ba Tư, yêu say đắm Rudaba, công chúa của Kabul. Khi Zal cầu hôn Rudaba, cha cô đã đưa ra một thách đố.

Ở Ba Tư có n thành phố, được gán nhãn từ 0 đến $n-1$, và m con đường hai chiều, được gán nhãn từ 0 đến $m-1$. Mỗi con đường nối hai thành phố phân biệt. Hai thành phố bất kỳ được nối bởi nhiều nhất một con đường. Một số con đường là *đường hoàng gia* được hoàng gia sử dụng để đi lại. Nhiệm vụ của Zal là xác định những con đường nào là con đường hoàng gia.

Zal có một bản đồ với tất cả các thành phố và những con đường ở Ba Tư. Anh ta không biết những con đường nào là con đường hoàng gia, nhưng anh ta có thể nhận được sự trợ giúp từ Simurgh, chú chim nhân từ huyền thoại bảo vệ Zal. Tuy nhiên, Simurgh không muốn tiết lộ trực tiếp tập các con đường hoàng gia. Thay vào đó, Simurgh nói với Zal rằng tập tất cả các con đường hoàng gia là một *bộ vàng*. Một tập các con đường là một bộ vàng khi và chỉ khi:

- Nó gồm *đúng* $n-1$ con đường, và
- Với mỗi cặp thành phố, có thể di chuyển từ thành phố này đến thành phố kia bằng cách di chuyển chỉ theo những con đường của tập này.

Ngoài ra, Zal có thể hỏi Simurgh một số câu hỏi. Với mỗi câu hỏi:

1. Zal chọn một tập các con đường là *bộ vàng*, và sau đó
2. Simurgh nói với Zal rằng có bao nhiêu con đường trong bộ vàng được chọn là con đường hoàng gia.

Chương trình của bạn cần giúp Zal tìm tập các con đường hoàng gia bằng cách hỏi Simurgh nhiều nhất q câu hỏi. Trình chấm sẽ đóng vai của Simurgh.

Chi tiết cài đặt

Bạn cần xây dựng thủ tục sau đây:

```
int[] find_roads(int n, int[] u, int[] v)
```

- n: số thành phố,
- u và v: mảng độ dài m. Với mọi $0 \leq i \leq m-1$, $u[i]$ và $v[i]$ là hai thành phố được nối bởi con đường i.
- Thủ tục này cần trả về một dãy độ dài $n-1$ chứa các nhãn của các con đường hoàng gia (theo thứ tự tùy ý).

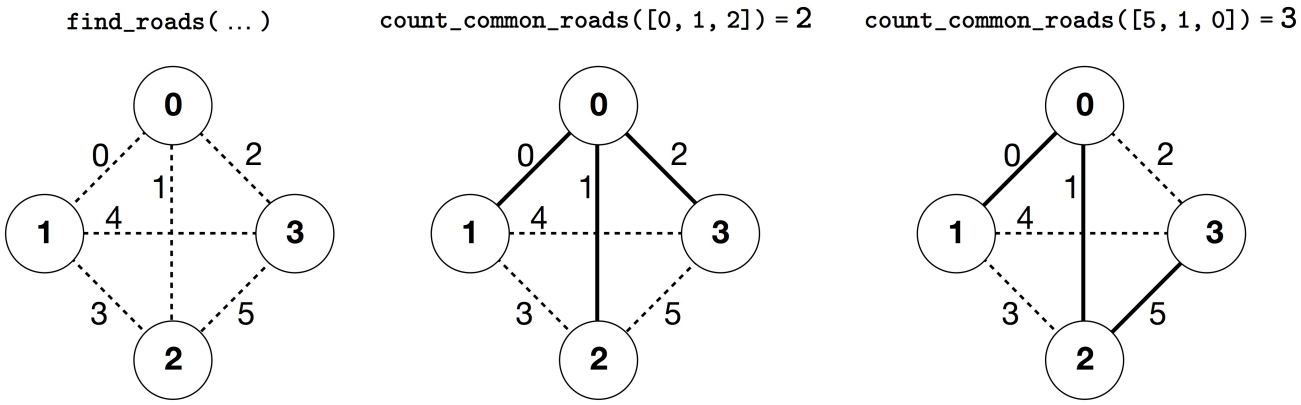
Giải pháp của bạn có thể thực hiện nhiều nhất q lần gọi đến thủ tục của trình chấm sau:

```
int count_common_roads(int[] r)
```

- r: mảng có độ dài $n-1$ chứa nhãn các con đường trong bộ vàng (theo thứ tự tùy ý).
- Thủ tục này trả về số lượng con đường hoàng gia trong r.

Ví dụ

```
find_roads(4, [0, 0, 0, 1, 1, 2], [1, 2, 3, 2, 3, 3])
```



Trong ví dụ này, có 4 thành phố và 6 con đường. Chúng ta ký hiệu (a, b) là con đường nối hai thành phố a và b. Các con đường được gán nhãn từ 0 đến 5 theo thứ tự sau: $(0, 1)$, $(0, 2)$, $(0, 3)$, $(1, 2)$, $(1, 3)$, và $(2, 3)$. Mỗi bộ vàng có $n - 1 = 3$ con đường.

Giả sử rằng các con đường hoàng gia là các con đường có nhãn 0, 1 và 5, nghĩa là các con đường $(0, 1)$, $(0, 2)$, và $(2, 3)$, thì:

- `count_common_roads([0, 1, 2])` trả về 2. Truy vấn này trên các con đường có nhãn $0, 1$ và 2, nghĩa là các con đường $(0, 1)$, $(0, 2)$ và $(0, 3)$. Hai trong số đó là các con đường hoàng gia.
- `count_common_roads([5, 1, 0])` trả về 3. Truy vấn này trên tập hợp tất cả các con đường hoàng gia.

Thủ tục `find_roads` cần trả về `[5, 1, 0]` hoặc bất kỳ mảng nào khác có độ dài 3 mà chứa ba phần tử đó.

Chú ý rằng các lệnh gọi sau đây không được phép:

- `count_common_roads([0, 1])`: ở đây độ dài của r không phải là 3.
- `count_common_roads([0, 1, 3])`: ở đây r không phải mô tả một bộ vàng, bởi vì không thể di chuyển từ thành phố 0 đến 3 chỉ sử dụng các con đường $(0, 1)$, $(0, 2)$, $(1, 2)$.

Ràng buộc

- $2 \leq n \leq 500$
- $n - 1 \leq m \leq n(n-1)/2$
- $0 \leq u[i], v[i] \leq n - 1$ (với mọi $0 \leq i \leq m - 1$)
- Với mọi $0 \leq i \leq m - 1$, con đường i nối hai thành phố khác nhau (nghĩa là $u[i] \neq v[i]$).
- 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.
- Tập tất cả các con đường hoàng gia là một bộ vàng.
- `find_roads` cần gọi `count_common_roads` nhiều nhất q lần. Trong mỗi lần gọi, tập hợp các con đường được chỉ định bởi r phải là một bộ vàng.

Subtasks

1. (13 điểm) $n \leq 7$, $q = 30\,000$
2. (17 điểm) $n \leq 50$, $q = 30\,000$
3. (21 điểm) $n \leq 240$, $q = 30\,000$
4. (19 điểm) $q = 12\,000$ và có một con đường giữa mọi cặp thành phố
5. (30 điểm) $q = 8\,000$

Trình chấm mẫu

Trình chấm mẫu đọc đầu vào theo khuôn dạng sau:

- dòng 1: n m
- dòng $2 + i$ (với mọi $0 \leq i \leq m - 1$): $u[i]$ $v[i]$
- dòng $2 + m$: $s[0]$ $s[1]$ $\ldots$ $s[n-2]$

Ở đây, $s[0], s[1], \ldots, s[n-2]$ là nhãn của các con đường hoàng gia.

Trình chấm mẫu đưa ra `YES`, nếu `find_roads` gọi `count_common_roads` nhiều nhất $30\,000$ lần, và trả về đúng tập các con đường hoàng gia. Ngược lại, nó đưa ra `NO`.

Lưu ý rằng thủ tục `count_common_roads` trong trình chấm mẫu không kiểm tra xem r có tất cả các thuộc tính của bộ vàng. Thay vào đó, nó đếm và trả về số lượng nhãn của các con đường hoàng gia trong mảng r. Tuy nhiên, nếu chương trình bạn nộp gọi `count_common_roads` với một tập các nhãn mà không mô tả một bộ vàng, thì thông báo chấm sẽ là 'Wrong Answer'.

Chú ý kỹ thuật

Thủ tục `count_common_roads` trong C ++ và Pascal sử dụng phương thức *pass by reference* vì các lý do hiệu quả. Bạn vẫn có thể gọi thủ tục theo cách thông thường. Trình chấm bảo đảm không thay đổi giá trị r.