

Sphinx's Riddle

Tượng Nhân Sư có một câu đố dành cho bạn. Bạn được cho một đồ thị có N đỉnh. Các đỉnh được đánh số từ 0 đến $N - 1$. Đồ thị có M cạnh, đánh số từ 0 đến $M - 1$. Mỗi cạnh kết nối hai chiều một cặp đỉnh phân biệt và là hai chiều. Cụ thể, đối với mỗi j từ 0 đến $M - 1$ (kể cả 0 và $M - 1$) cạnh j nối các đỉnh $X[j]$ và $Y[j]$. Có nhiều nhất một cạnh nối một cặp đỉnh bất kỳ. Hai đỉnh được gọi là **kề nhau** nếu chúng được nối với nhau bởi một cạnh.

Một dãy các đỉnh v_0, v_1, \dots, v_k (với $k \geq 0$) được gọi là một **đường đi** nếu mỗi hai đỉnh liên tiếp v_l và v_{l+1} (với mỗi l sao cho $0 \leq l < k$) là kề nhau. Ta nói rằng đường đi v_0, v_1, \dots, v_k **kết nối** các đỉnh v_0 và v_k . Trong đồ thị đã cho, mỗi cặp đỉnh được kết nối bởi một đường đi nào đó.

Có $N + 1$ màu, được đánh số từ 0 đến N . Màu N là màu đặc biệt và được gọi là **màu của Nhân Sư**. Mỗi đỉnh được gán một màu. Cụ thể, đỉnh i ($0 \leq i < N$) có màu $C[i]$. Có thể có nhiều đỉnh cùng màu nhau, và có thể có những màu không được gán cho bất kỳ đỉnh nào. Không có đỉnh nào có màu của Nhân Sư, nghĩa là $0 \leq C[i] < N$ ($0 \leq i < N$).

Đường đi v_0, v_1, \dots, v_k (với $k \geq 0$) được gọi là **đơn sắc** nếu như tất cả các đỉnh của nó có cùng màu, tức là $C[v_l] = C[v_{l+1}]$ (với mỗi l thỏa mãn $0 \leq l < k$). Ngoài ra, chúng ta nói rằng các đỉnh p và q ($0 \leq p < N$, $0 \leq q < N$) có cùng **thành phần đơn sắc** khi và chỉ khi chúng được kết nối bởi một đường đi đơn sắc.

Bạn biết các đỉnh và cạnh, nhưng bạn không biết mỗi đỉnh có màu gì. Bạn muốn tìm ra màu sắc của các đỉnh, bằng cách thực hiện **thí nghiệm tô màu lại**.

Trong một thí nghiệm tô màu lại, bạn có thể tô màu lại nhiều đỉnh tùy ý. Cụ thể, để thực hiện một thí nghiệm tô màu lại, đầu tiên bạn chọn một mảng E có kích thước N , trong đó đối với mỗi i ($0 \leq i < N$), $E[i]$ nằm giữa -1 và N **bao gồm cả -1 và N** . Sau đó, màu của mỗi đỉnh i trở thành $S[i]$, trong đó giá trị của $S[i]$ là:

- $C[i]$, tức là màu gốc của đỉnh i , nếu $E[i] = -1$, hoặc
- $E[i]$, trái lại.

Lưu ý điều này có nghĩa là bạn có thể sử dụng màu của Nhân Sư khi tô màu lại.

Cuối cùng, Tượng Nhân Sư công bố số lượng các thành phần đơn sắc trong đồ thị, sau khi thiết lập màu của mỗi đỉnh i thành $S[i]$ ($0 \leq i < N$). Việc tô màu mới chỉ được áp dụng cho thí nghiệm tô màu lại này, vì vậy **màu của tất cả các đỉnh sẽ trở về màu ban đầu sau khi thí nghiệm kết thúc**.

Nhiệm vụ của bạn là xác định màu của các đỉnh trong đồ thị bằng cách thực hiện tối đa 2 750 thí nghiệm tô màu lại. Bạn cũng có thể nhận được một phần điểm nếu bạn xác định đúng cho mọi cặp đỉnh kề nhau liệu chúng có cùng màu hay không.

Chi tiết cài đặt

Bạn cần cài đặt hàm sau.

```
std::vector<int> find_colours(int N,  
                             std::vector<int> X, std::vector<int> Y)
```

- N : số lượng đỉnh trong đồ thị.
- X, Y : các mảng độ dài M mô tả các cạnh.
- Hàm này cần trả về một mảng G độ dài N , biểu diễn màu của các đỉnh trong đồ thị.
- Hàm này được gọi đúng một lần cho mỗi trường hợp test.

Hàm trên có thể thực hiện các lời gọi đến hàm sau để thực hiện các thí nghiệm tô màu lại:

```
int perform_experiment(std::vector<int> E)
```

- E : mảng độ dài N chỉ ra các đỉnh cần được tô màu lại thế nào.
- Hàm này trả về số lượng các thành phần đơn sắc sau khi tô lại các đỉnh theo E .
- Hàm này có thể được gọi tối đa 2 750 lần.

Trình chấm là **không thích ứng**, nghĩa là, màu của các đỉnh được cố định trước khi thực hiện một lời gọi `find_colours`.

Các ràng buộc

- $2 \leq N \leq 250$
- $N - 1 \leq M \leq \frac{N \cdot (N-1)}{2}$
- $0 \leq X[j] < Y[j] < N$ với mỗi j thoả mãn $0 \leq j < M$.
- $X[j] \neq X[k]$ hoặc $Y[j] \neq Y[k]$ với mỗi j và k thoả mãn $0 \leq j < k < M$.
- Mỗi cặp đỉnh được kết nối bởi đường đi nào đó.
- $0 \leq C[i] < N$ với mỗi i thoả mãn $0 \leq i < N$.

Các subtask

Subtask	Điểm	Ràng buộc thêm
1	3	$N = 2$
2	7	$N \leq 50$
3	33	Đồ thị là một đường đi: $M = N - 1$ và các đỉnh j và $j + 1$ kề nhau ($0 \leq j < M$).
4	21	Đồ thị đầy đủ: $M = \frac{N \cdot (N - 1)}{2}$ và hai đỉnh bất kỳ là kề nhau.
5	36	Không có ràng buộc nào thêm.

Trong mỗi subtask, bạn có thể nhận được một phần điểm nếu chương trình của bạn xác định đúng cho mỗi cặp đỉnh kề nhau liệu chúng có cùng màu hay không.

Cụ thể, bạn nhận được toàn bộ điểm của một subtask nếu trong tất cả các trường hợp test của subtask đó, mảng G được trả về bởi `find_colours` hoàn toàn giống với mảng C (tức là $G[i] = C[i]$ với mọi i thỏa mãn $0 \leq i < N$). Trái lại, bạn nhận được 50% số điểm đối với một subtask nếu các điều kiện sau đây thỏa mãn trong tất cả các trường hợp test của subtask đó:

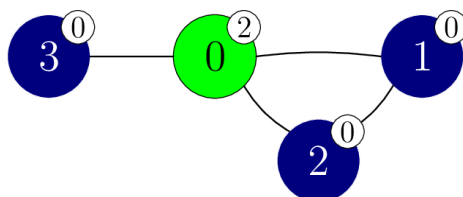
- $0 \leq G[i] < N$ với mỗi i thỏa mãn $0 \leq i < N$;
- Với mỗi j thỏa mãn $0 \leq j < M$:
 - $G[X[j]] = G[Y[j]]$ khi và chỉ khi $C[X[j]] = C[Y[j]]$.

Ví dụ

Xét lời gọi sau.

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

Đối với ví dụ này, giả sử rằng màu (ẩn) của các đỉnh được đưa ra bởi $C = [2, 0, 0, 0]$. Tình huống này được thể hiện trong hình sau. Màu cũng được biểu thị bằng các con số trên nhãn màu trắng gắn ở mỗi đỉnh.



Hàm này có thể gọi `perform_experiment` như sau.

```
perform_experiment([-1, -1, -1, -1])
```

Trong lời gọi này, không có đỉnh nào được tô màu lại vì tất cả các đỉnh đều giữ nguyên màu gốc.

Xét đỉnh 1 và đỉnh 2. Cả hai đều có màu 0 và đường đi 1, 2 là đường đi đơn sắc. Kết quả là, các đỉnh 1 và 2 nằm trong cùng một thành phần đơn sắc.

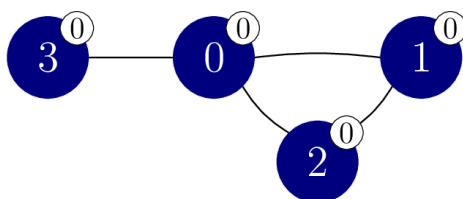
Xét đỉnh 1 và đỉnh 3. Mặc dù cả hai đều có màu 0, chúng có các thành phần đơn sắc khác nhau vì không có đường đi đơn sắc nào kết nối chúng.

Như vậy, có 3 thành phần đơn sắc, với các đỉnh $\{0\}$, $\{1, 2\}$ và $\{3\}$. Do đó, lời gọi này trả về 3.

Bây giờ hàm có thể gọi `perform_experiment` như sau.

```
perform_experiment([0, -1, -1, -1])
```

Trong lời gọi này, chỉ có đỉnh 0 được đổi màu thành màu 0, do đó, việc tô màu như trong hình sau.

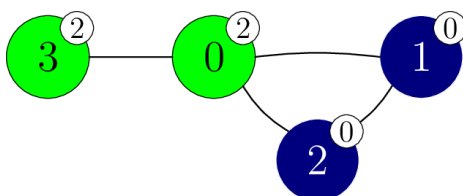


Lời gọi này trả về 1 vì tất cả các đỉnh đều thuộc cùng một thành phần đơn sắc. Bây giờ ta có thể suy ra rằng các đỉnh 1, 2 và 3 có màu 0.

Do đó, hàm có thể gọi `perform_experiment` như sau.

```
perform_experiment([-1, -1, -1, 2])
```

Trong lời gọi này, đỉnh 3 được tô màu lại thành màu 2, do đó, việc tô màu như trong hình sau.



Lời gọi này trả về 2 vì có 2 thành phần đơn sắc, tương ứng với các đỉnh $\{0, 3\}$ và $\{1, 2\}$. Ta có thể suy ra rằng đỉnh 0 có màu 2.

Do đó, hàm `find_colours` trả về mảng $[2, 0, 0, 0]$. Vì $C = [2, 0, 0, 0]$ nên kết quả đạt điểm tối đa.

Lưu ý rằng cũng có nhiều giá trị trả về để đạt được 50% số điểm, ví dụ $[1, 2, 2, 2]$ hoặc $[1, 2, 2, 3]$.

Trình chấm mẫu

Định dạng dữ liệu vào:

```
N  M
C[0]  C[1]  ...  C[N-1]
X[0]  Y[0]
X[1]  Y[1]
...
X[M-1]  Y[M-1]
```

Định dạng kết quả ra:

```
L  Q
G[0]  G[1]  ...  G[L-1]
```

Trong đó, L là độ dài của mảng G trả về bởi `find_colours`, và Q là số lượng lời gọi tới `perform_experiment`.