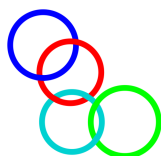


Các vòng móc dù

Một phiên bản rất tinh vi và có từ rất sớm của cái mà ngày nay ta gọi là cái dù được mô tả trong cuốn sách *Codex Atlanticus* của Leonardo (khoảng 1485). Dù của Leonardo được cấu thành từ vải lanh kín được căng bởi một cấu trúc bằng gỗ có dạng kim tự tháp.

Các vòng móc nối

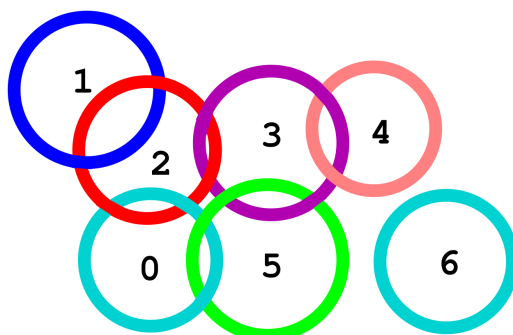
Sau đó hơn 500 năm, nhà nhảy dù Adrian Nicholas đã kiểm thử thiết kế dù của Leonardo. Để thực hiện việc này, một cấu trúc nhẹ hiện đại gắn liền dù của Leonardo với thân thể con người. Chúng ta muốn sử dụng các vòng móc nối như là các móc cho miếng vải lanh kín. Mỗi vòng được làm từ vật liệu rất bền và dễ uốn. Các vòng dễ dàng được móc nối với nhau bởi vì mỗi vòng có thể được mở ra và lại được đóng kín lại. Một cấu hình đặc biệt của các vòng được móc nối với nhau là *chuỗi*. *Chuỗi* là một dãy các vòng, trong đó mỗi vòng được móc nối với (nhiều nhất là hai) vòng láng giềng của nó, như được chỉ ra trong hình vẽ dưới đây. Dãy này phải có vòng bắt đầu và vòng kết thúc (các vòng được móc nối với đúng một vòng khác). Nói riêng, một vòng đơn lẻ cũng là chuỗi.



Rõ ràng còn có các cấu hình khác, bởi vì một vòng có thể được móc nối với ba hoặc nhiều hơn vòng khác. Chúng ta nói rằng một vòng là *then chốt* nếu sau khi mở và loại bỏ nó tất cả các vòng còn lại tạo thành một tập các chuỗi (hoặc không còn lại một vòng nào). Nói cách khác, không còn cái gì khác ngoài các chuỗi.

Ví dụ

Xét 7 vòng được đánh số từ 0 đến 6 trong hình vẽ tiếp theo. Có hai vòng là then chốt. Một vòng then chốt là vòng 2: sau khi loại bỏ vòng này, các vòng còn lại tạo thành các chuỗi [1], [0, 5, 3, 4] và [6]. Một vòng then chốt khác là vòng 3: sau khi loại bỏ vòng này, các vòng còn lại tạo thành các chuỗi [1, 2, 0, 5], [4] và [6]. Nếu như loại bỏ bất cứ vòng nào khác ta không thu được tập các chuỗi rời nhau. Chẳng hạn, sau khi loại bỏ vòng 5: mặc dù ta có vòng 6 là một chuỗi, nhưng các vòng móc nối với nhau 0, 1, 2, 3 và 4 không tạo thành chuỗi.



Phát biểu bài toán

Nhiệm vụ của bạn là đếm số vòng then chốt trong một cấu hình cho trước được truyền tải cho chương trình của bạn.

Tại thời điểm bắt đầu, có một số vòng rời nhau. Sau đó, các vòng được móc nối với nhau. Tại một thời điểm cho trước bạn sẽ được yêu cầu đếm số vòng then chốt trong cấu hình hiện tại. Cụ thể, bạn phải cài đặt ba chương trình con.

- `Init(N)` — chương trình này được gọi đúng một lần khi bắt đầu để nhận được N vòng rời nhau được đánh số từ 0 đến $N-1$ (kể cả hai đầu) trong trạng thái xuất phát.
- `Link(A, B)` — hai vòng được đánh số là A và B được móc nối với nhau. Bạn được đảm bảo rằng A và B là khác nhau và chưa được móc nối với nhau; ngoài điều kiện này, không có điều kiện bổ sung nào đối với A và B , nói riêng không có điều kiện nào nảy sinh từ các ràng buộc vật lý. Rõ ràng, `Link(A, B)` và `Link(B, A)` là tương đương.
- `CountCritical()` — trả lại số lượng vòng then chốt đối với cấu hình các vòng móc nối hiện tại.

Ví dụ

Xét cấu hình với $N=7$ vòng và giả sử rằng ở lúc ban đầu chúng không móc nối với nhau. Chúng ta xét một dãy các lệnh gọi có thể, trong đó sau lệnh gọi cuối cùng ta thu được tình trạng được mô tả trong hình vẽ.

Lệnh gọi	Giá trị trả lại
Init(7)	
CountCritical()	7
Link(1, 2)	
CountCritical()	7
Link(0, 5)	
CountCritical()	7
Link(2, 0)	
CountCritical()	7
Link(3, 2)	
CountCritical()	4
Link(3, 5)	
CountCritical()	3
Link(4, 3)	
CountCritical()	2

Subtask 1 [20 points]

- $N \leq 5\,000$.
- Hàm `CountCritical` được gọi chỉ một lần, sau tất cả các lệnh gọi khác; hàm `Link` được gọi không quá 5 000 lần.

Subtask 2 [17 points]

- $N \leq 1\,000\,000$.
- Hàm `CountCritical` được gọi chỉ một lần, sau tất cả các lệnh gọi khác; hàm `Link` được gọi không quá 1 000 000 lần.

Subtask 3 [18 points]

- $N \leq 20\,000$.
- Hàm `CountCritical` được gọi không quá 100 lần; hàm `Link` được gọi không quá 10000 lần.

Subtask 4 [14 points]

- $N \leq 100\,000$.
- Hàm `CountCritical` và `Link` được gọi tổng cộng không quá 100 000 lần.

Subtask 5 [31 points]

- $N \leq 1\,000\,000$.

- Hàm `CountCritical` và hàm `Link` được gọi tổng cộng không quá 1 000 000 lần.

Chi tiết cài đặt

Bạn phải nộp đúng một file gọi là `rings.c`, `rings.cpp` hoặc `rings.pas`. File này cài đặt các chương trình con đã được mô tả ở trên sử dụng các mẫu sau đây.

C/C++ programs

```
void Init(int N);
void Link(int A, int B);
int CountCritical();
```

Pascal programs

```
procedure Init(N : LongInt);
procedure Link(A, B : LongInt);
function CountCritical() : LongInt;
```

Các chương trình con này phải hoạt động như mô tả ở trên. Tất nhiên bạn được tự do cài đặt thêm các chương trình con khác để sử dụng trong nội bộ nó. Các lần giao nộp của bạn không được giao tiếp dưới bất cứ hình thức nào với vào/ra chuẩn, cũng như với bất cứ file nào khác.

Sample grader

Chương trình grader sẽ đọc dữ liệu vào theo khuôn dạng:

- dòng 1: `N, L`;
- dòng 2, ..., `L + 1`:
 - -1 để thực hiện lệnh gọi `CountCritical`;
 - `A, B` là các thông số cho `Link`.

Chương trình grader sẽ in ra tất cả các kết quả từ `CountCritical`.