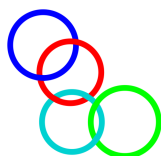


Anelli del paracadute

Un antenato piuttosto sofisticato di quello che oggi chiamiamo paracadute viene descritta nel *Codex Atlanticus* di Leonardo (ca. 1485). Il paracadute di Leonardo era composto da un telo di lino cerato, tenuto aperto da una struttura piramidale di legno.

Anelli agganciati

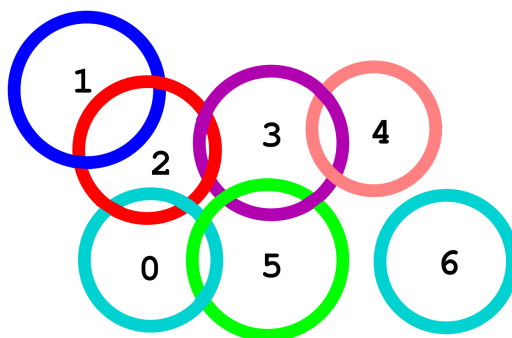
Il paracadutista Adrian Nicholas ha provato il progetto di Leonardo più di 500 anni dopo. Per farlo, ha collegato il paracadute di Leonardo al corpo con una struttura moderna e leggera. Vogliamo usare anelli agganciati, che forniscano anche ganci per il telo di lino cerato. Ciascun anello è composto di un materiale flessibile e resistente. Gli anelli possono essere agganciati l'uno all'altro facilmente, visto che ciascun anello può essere aperto e richiuso. La *catena* è una sequenza di anelli in cui ogni anello è collegato agli anelli adiacenti (che sono al più due), come illustrato sotto. Questa sequenza deve avere un inizio ed una fine (ovvero anelli che sono connessi ad un unico anello ciascuno). Come caso particolare, anche un singolo anello è una catena.



Sono ovviamente possibili altre configurazioni, poiché si può agganciare un anello a tre o più anelli. Un anello si dice *critico* se dopo averlo aperto e rimosso, tutti gli anelli rimanenti formano un'insieme di catene (o se non rimangono altri anelli). In altre parole, devono rimanere soltanto catene.

Esempio

Considera i 7 anelli della figura seguente, numerati da 0 a 6. Ci sono due anelli critici. 2 è un anello critico: dopo la sua rimozione, gli anelli rimanenti formano le catene [1], [0, 5, 3, 4] e [6]. L'altro anello critico è 3: dopo la sua rimozione, gli anelli rimanenti formano le catene [1, 2, 0, 5], [4] e [6]. Se si rimuove un qualunque altro anello, non si ottiene un insieme di catene disgiunte. Per esempio, dopo la rimozione di 5: nonostante [6] sia una catena, gli anelli 0, 1, 2, 3 e 4 sono collegati ma non formano una catena.



Descrizione del problema

Il tuo compito è quello di contare il numero di anelli critici in una configurazione data, che verrà comunicata al tuo programma.

Inizialmente c'è un certo numero di anelli disgiunti. Successivamente gli anelli vengono agganciati. In un qualunque momento, ti si può chiedere di restituire il numero di anelli critici nella configurazione corrente. In particolare, devi implementare tre funzioni.

- `Init(N)` — viene chiamata esattamente una volta all'inizio per comunicare che nella configurazione iniziale ci sono N anelli disgiunti numerati da 0 ad $N - 1$ (inclusi).
- `Link(A, B)` — i due anelli numerati A e B vengono collegati. È garantito che A e B siano differenti e che non siano ancora collegati direttamente; per il resto, non ci sono ulteriori condizioni su A e B , in particolare non ci sono condizioni dovute a limiti fisici. Ovviamente `Link(A, B)` e `Link(B, A)` sono equivalenti.
- `CountCritical()` — restituisce il numero di anelli critici per la configurazione corrente di anelli collegati.

Esempio

Considera la figura con $N = 7$ anelli e supponi che siano inizialmente scollegati. Ecco una possibile sequenza di chiamate, alla fine della quale si ottiene la situazione rappresentata in figura.

Chiamata	Risultato
<code>Init(7)</code>	
<code>CountCritical()</code>	7
<code>Link(1, 2)</code>	
<code>CountCritical()</code>	7
<code>Link(0, 5)</code>	
<code>CountCritical()</code>	7
<code>Link(2, 0)</code>	
<code>CountCritical()</code>	7
<code>Link(3, 2)</code>	
<code>CountCritical()</code>	4
<code>Link(3, 5)</code>	
<code>CountCritical()</code>	3
<code>Link(4, 3)</code>	
<code>CountCritical()</code>	2

Subtask 1 [20 punti]

- $N \leq 5\,000$.
- La funzione `CountCritical` viene chiamata solo una volta, dopo tutte le altre chiamate; la funzione `Link` viene chiamata al più 5 000 volte.

Subtask 2 [17 punti]

- $N \leq 1\,000\,000$.
- La funzione `CountCritical` viene chiamata solo una volta, dopo tutte le altre chiamate; la funzione `Link` viene chiamata al più 1 000 000 di volte.

Subtask 3 [18 punti]

- $N \leq 20\,000$.
- La funzione `CountCritical` viene chiamata al più 100 volte; la funzione `Link` viene chiamata al più 10 000 volte.

Subtask 4 [14 punti]

- $N \leq 100\,000$.
- Le funzioni `CountCritical` e `Link` vengono chiamate, in totale, al più 100 000 volte.

Subtask 5 [31 punti]

- $N \leq 1\,000\,000$.
- Le funzioni `CountCritical` e `Link` vengono chiamate, in totale, al più 1 000 000 di volte.

Dettagli implementativi

Devi inviare esattamente un file, chiamato `rings.c`, `rings.cpp` oppure `rings.pas`. Questo file deve implementare le funzioni sopra descritte utilizzando i seguenti prototipi.

Programmi C/C++

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

Programmi Pascal

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

Queste funzioni devono comportarsi come descritto sopra. Ovviamente sei libero di implementare altre funzioni per uso interno. Le tue sottoposizioni non devono interagire in nessun modo con l'input/output standard né con nessun altro file.

Grader di esempio

Il grader di esempio legge l'input nel seguente formato:

- linea 1: N, L;
- linee 2, ..., L + 1:
 - -1 per chiamare `CountCritical`;
 - i parametri A, B per `Link`.

Il grader di esempio stamperà tutti i risultati di `CountCritical`.