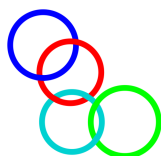


Padobranski prstenovi

Jedna ranija, ali svejedno sofisticirana verzija današnjeg padobrana opisana je u Leonardovoj *Codex Atlanticus* (ca. 1485). Leonardov padobran se sastojao od zašivenih lanenih plahti koje je otvorenim držala piramidalna drvena struktura.

Povezani prstenovi

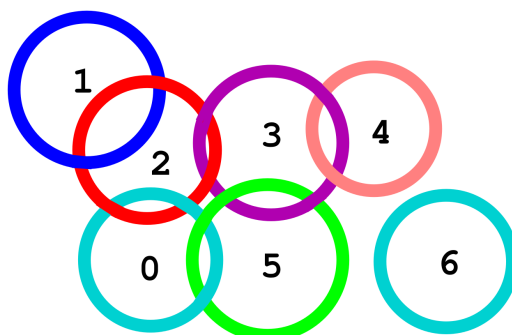
Padobranac Adrian Nicholas je testirao Leonardov dizajn više od 500 godina kasnije. Za ovo je korištena moderna struktura koja bi vezala Leonardov padobran za ljudsko tijelo. Mi želimo koristiti vezane prstenove, koji također imaju kvačice za lanene plahte. Prstenovi su fleksibilni i lako se vežu pošto se mogu otvoriti i zatvoriti po potrebi. Posebna konfiguracija vezanih prstenova se naziva *lanac*. *Lanac* je niz od jedan ili više prstenova povezanih tako da je svaki prsten može biti vezan za (najviše) dva susjedna prstena, kao što je prikazano na slici dolje. Taj niz mora imati početak i kraj (dva prstena koja su povezana za najviše jedan drugi prsten). Također, nepovezan prsten je također lanac.



Druge konfiguracije su očito moguće, pošto neki prsten može biti vezan za tri ili više drugih prstena. Kažemo da je neki prsten *kritičan* ako nakon otvaranja i uklanjanja tog prstena, ostali prstenovi čine skup disjunktnih lanaca (ili nema drugih prstenova).

Primjer

Uzmite u obzir sedam prstenova u sljedećoj slici, numeriranih od 0 do 6. Postoje dva kritična prstena. Jedan je 2: ako bi ga uklonili, preostali bi lanci [1], [0, 5, 3, 4] i [6]. Drugi kritični prsten je 3: ako ga uklonimo, ostali prsteni čine lance [1, 2, 0, 5], [4] i [6]. Ako bi uklonili bilo koji drugi prsten, ne bi dobili skup disjunktnih lanaca. Npr. ako uklonimo prsten 5 dobijemo lanac [6], ali prsteni 0, 1, 2, 3 i 4 ne čine lanac.



Zadatak

Vaš zadatak je izbrojati broj kritičnih prstena u konfiguracijama u određenim trenucima.

Na početku postoji određen broj međusobno odvojenih prstenova. Nakon toga, prsteni se spajaju i u određenim trenucima se traži da izračunate broj kritičnih prstenova. Morate implementirati sljedeća tri potprograma.

- `Init(N)` — poziva se točno jednom na početku programa i označava da postoji N odvojenih prstenova numeriranih od 0 do $N - 1$ (uključivo) u početnoj konfiguraciji.
- `Link(A, B)` — označava spajanje prstenova označenih s A i B . Garantirano je da će A i B biti različiti i prethodno nespojeni (direktno). Osim toga, nema nikakvih ograničenja na A i B , primjerice nekih koje bi mogle biti posljedica fizičkih ograničenja. Očito, `Link(A, B)` istovjetno je s `Link(B, A)`.
- `CountCritical()` — vraća broj kritičnih prstenova u trenutnoj konfiguraciji.

Primjer

Promotrite primjer $n = 7$ prstenova i pretpostavite da su na početku svi odvojeni. Slijedi prikaz mogućeg niza poziva, gdje nakon zadnjeg poziva imamo situaciju kao na gornjoj slici.

Poziv	Povratna vrijednost
<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

Podzadatak 1 [20 bodova]

- $N \leq 5\,000$.
- Funkcija `CountCritical` će biti pozvana samo jednom, nakon svih ostalih poziva; funkcija `Link` će biti pozvana najviše 5 000 puta.

Podzadatak 2 [17 bodova]

- $N \leq 1\,000\,000$.
- Funkcija `CountCritical` će biti pozvana samo jednom, nakon svih ostalih poziva; funkcija `Link` će biti pozvana najviše 1 000 000 puta.

Podzadatak 3 [18 bodova]

- $N \leq 20\,000$.
- Funkcija `CountCritical` će biti pozvana najviše 100 puta; funkcija `Link` će biti pozvana najviše 10 000 puta.

Podzadatak 4 [14 bodova]

- $N \leq 100\,000$.
- Funkcije `CountCritical` i `Link` će biti pozvane ukupno najviše 100 000 puta.

Podzadatak 5 [31 bodova]

- $N \leq 1\,000\,000$.
- Funkcije `CountCritical` i `Link` će biti pozvane ukupno najviše 1 000 000 puta.

Implementacija

Morate poslati samo jednu datoteku, nazvanu `rings.c`, `rings.cpp` ili `rings.pas`. Ta datoteka mora implementi tri gore opisane funkcije koristeći sljedeće deklaracije.

C/C++ programi

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

Pascal programi

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

Ovi potprogrami se moraju ponašati prema gore opisanome. Naravno, za vaše potrebe možete implementirati druge pomoćne potprograme. Imajte na umu da vaše rješenje ne smije koristiti standardni ulaz te izlaz, kao i bilo kakve druge datoteke.

Primjer biblioteke za testiranje

Biblioteka za testiranje očekuje sljedeći ulazni format:

- linija 1: N, L;
- linije 2, ..., L + 1:
 - -1 za poziv `CountCritical`;
 - A, B kao parametri za poziv `Link`.

Biblioteka će ispisati sve rezultate koje vrate pozivi iz `CountCritical`.