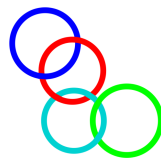


Parachute rings

Sebuah versi lama dan cukup rumit dari benda yang sekarang kita namakan parasut dijelaskan dalam Codex Atlanticus dari Leonardo (tahun 1485). Parasut Leonardo terdiri dari kain linen kedap-air terbentang dan dikaitkan pada sebuah kayu berbentuk piramid.

Linked rings

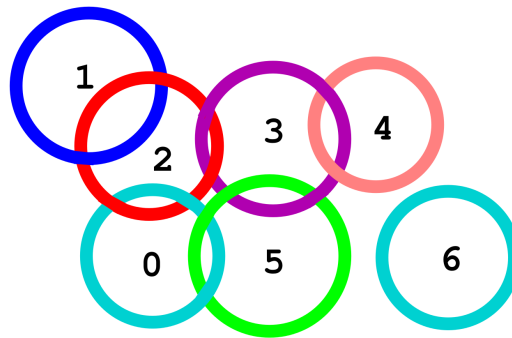
Adrian Nicholas, seorang penerbang layang menguji desain Leonardo 500 tahun yang lalu. Untuk ini, sebuah struktur modern yang ringan dipakai untuk mengikat parasut Leonardo ke tubuh manusia. Untuk ini kita ingin menggunakan cincin yang saling berkaitan, yang juga membentuk hooks untuk kain linen kedap-air. Setiap cincin terbuat dari material yang fleksibel dan kuat. Cincin dapat dengan mudah dikaitkan bersama karena setiap cincin dapat dibuka dan ditutup kembali. Sebuah konfigurasi khusus dari cincin berkait disebut *rantai*. Sebuah *rantai* adalah sebuah sekuens (rangkai) cincin-cincin yang mana setiap cincin hanya terhubung dengan (paling banyak dua) tetangganya, seperti yang diilustrasikan di bawah ini. Sekuens ini harus memiliki sebuah awal dan sebuah akhir (cincin-cincin yang masing-masing terhubung ke paling banyak sebuah cincin). Lebih spesifik lagi, sebuah cincin juga merupakan sebuah rantai.



Jelas bahwa mungkin juga ada konfigurasi yang lain, sebab sebuah cincin dapat dikaitkan ke tiga atau lebih cincin. Kita katakan bahwa sebuah cincin adalah *kritikal* jika setelah membuka dan melepaskannya, semua cincin tersisa membentuk himpunan disjoint rantai-rantai (atau tak ada cincin yang tersisa). Dengan kata lain, bisa tidak ada rantai tersisa.

Contoh

Perhatikan 7 cincin pada gambar di bawah, diberi nomor mulai dari 0 sampai dengan 6. Ada dua cincin kritikal. Salah satu cincin kritikal adalah 2: setelah dilepaskan, sisa cincin membentuk rantai [1], [0,5,3,4] dan [6]. Cincin kritikal lainnya adalah 3 : setelah dilepaskan, sisa cincin membentuk rantai [1,2,0,5], [4] dan [6]. Jika kita melepaskan salah satu cincin yang lain, kita tidak memperoleh sebuah himpunan rantai yang disjoint. Misalnya, setelah melepaskan cincin 5, walaupun [6] adalah sebuah rantai, cincin terkait 0,1,2,3 dan 4 tidak membentuk sebuah rantai.



Statement

Tugas Anda adalah menghitung banyaknya cincin kritisal dalam sebuah konfigurasi yang diberikan, yang akan dikomunikasikan ke program Anda.

Pada awalnya, ada sejumlah cincin terpisah. Setelah itu, cincin dikaitkan bersama. Pada setiap saat, anda dapat diminta untuk mengembalikan banyaknya cincin kritisal pada konfigurasi saat itu. Tepatnya, anda harus mengimplementasi 3 subrutin

- `Init(N)` — dipanggil satu kali saja pada wal program, menyatakan bahwa ada N disjoint cincin yang dinomori dari 0 s.d. $N - 1$ (inklusif) pada konfigurasi awal.
- `Link(A, B)` — dua buah cincin A dan B dikaitkan satu sama lain. Dijamin bahwa A dan B berbeda dan belum dikaitkan secara langsung; selain itu, tak ada kondisi tambahan pada A dan B , khususnya tak ada konstrain fisik lainnya. Jelas, bahwa `Link(A, B)` dan `Link(B, A)` adalah ekuivalen
- `CountCritical()` —mengembalikan banyaknya cincin kritisal pada konfigurasi cincin berkait saat itu

Example

Tinjau gambar dengan $N = 7$ cincin dan misalkan bahwa cincin-cincin tersebut pada mulanya tak terkait. Ditunjukkan sebuah kemungkinan sederetan call, di mana setelah call yang terakhir kita memperoleh situasi seperti ditunjukkan pada gambar.

Call	Returns
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]

- $N \leq 5\,000$.
- Fungsi `CountCritical` dipanggil hanya satu kali, setelah call lainnya; fungsi `Link` dipanggil paling banyak 5 000 kali.

Subtask 2 [17 points]

- $N \leq 1\,000\,000$.
- The function `CountCritical` is called only once, after all the other calls; the function `Link` is called at most 1 000 000 times.

Subtask 3 [18 points]

- $N \leq 20\,000$.
- The function `CountCritical` is called at most 100 times; the function `Link` is called at most 10 000 times.

Subtask 4 [14 points]

- $N \leq 100\,000$.
- The functions `CountCritical` and `Link` are called, in total, at most 100 000 times.

Subtask 5 [31 points]

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

- The functions `CountCritical` and `Link` are called, in total, at most 1 000 000 times.

Implementation details

You have to submit exactly one file, called `rings.c`, `rings.cpp` or `rings.pas`. This file implements the subprograms described above using the following signatures.

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;
```

Subprogram harus berperilaku seperti yang dijelaskan di atas. Selain itu, anda bebas untuk mengimplementasikan subprogram lain untuk kepentingan internal. Program yang Anda kumpulkan dilarang berinteraksi dengan input/output standar, atau dengan file lain.

Sample grader

Contoh grader membaca input dalam format sebagai berikut:

- baris 1: N, L ;
- baris 2, ..., $L + 1$:
 - -1 to invoke `CountCritical`;
 - A, B parameters to `Link`.

Contoh grader akan mencetak semua hasil dari `CountCritical`.