International Olympiad in Informatics 2012
23-30 September 2012
Sirmione - Montichiari, Italy
Competition tasks, day 1: Leonardo's inventions and projects

rings
Latvian — 1.2

Izpletņa gredzeni

Agrīna un diezgan izsmalcināta izpletņa versija ir aprakstīta Leonardo da Vinči 1485.gada darbā *Codex Atlanticus*. Leonardo izpletni veidoja cieši savienots lina audums, kuru vaļējā stāvoklī notur piramīdas formas koka karkass.

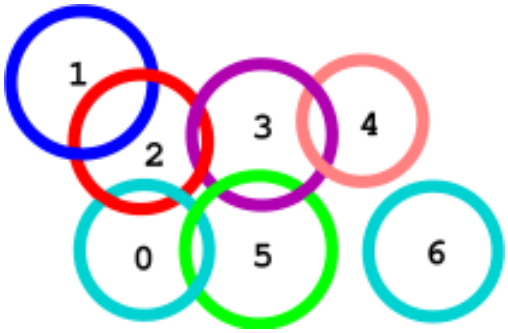
Savienotie gredzeni

Izpletņlēcējs Adrians Nikolass izmēģināja Leonardo izgudrojumu pēc vairāk nekā 500 gadiem. Leonardo izpletnis ar vieglas mūsdienīgas konstrukcijas palīdzību tika piesaistīts izpletņlēcēja ķermenim. Ir vēlme konstrukcijā izmantot savienotus gredzenus, kas ļautu piekabināt cieši savienoto lina drēbi. Katrs gredzens ir neliela, lokana un izturīga materiāla karabīne. Gredzenus var viegli savienot savā starpā tā, ka katrs gredzens var tikt viegli aiztaisīts vai atvērts. Speciālu savienotu gredzenu virkni sauc par *ķēdi*: viena vai vairāku gredzenu virkni, kurā katrs gredzens, izņemot pirmo un pēdējo, ir savienots ar tieši diviem citiem gredzeniem. Pie kam ķēdei ir jābūt sākumam un beigām - tādiem gredzeniem, kas savienoti ar tieši vienu citu gredzenu. Ķēdes speciālgadījums ir viens atsevišķs gredzens. Ķēdes piemērs ir dots zemāk redzamajā zīmējumā.

Tā kā kāds gredzens var būt savienots arī ar trim vai vairāk gredzeniem, tad ir iespējmas arī citas konfigurācijas. Gredzenu sauksim par *kritisku*, ja pēc tā atvēršanas un izņemšanas visi atlikušie gredzeni veido ķēžu kopu (vai arī vairs nav atlicis neviens gredzens). Citiem vārdiem sakot, nav nevienas tādas gredzenu konfigurācijas, kas nebūtu ķēde.

Piemērs

Aplūkosim nākamajā zīmējumā dotos septiņus gredzenus, kas sanumurēti ar skaitļiem no 0 līdz 6. Ir divi kritiskie gredzeni: 2 (pēc tā atvēršanas un izņemšanas atlikušie gredzeni veido ķēdes [1], [0, 5, 3, 4] un [6]) un 3 (pēc tā atvēršanas un izņemšanas atlikušie gredzeni veido ķēdes [1, 2, 0, 5], [4] un [6]). Ja tiktu atvērts un izņemts jebkurš cits gredzens, atlikušie gredzeni neveidos nešķeļošos ķēžu kopu. Piemēram, atverot un izņemot 5. gredzenu, mēs iegūstam ķēdi [6], bet pārējie gredzeni 0, 1, 2, 3 un 4 ķēdi neveido.



Uzdevums

Jūsu uzdevums ir noteikt kritisko gredzenu skaitu konfigurācijā, kas tiks paziņota jūsu programmai.

Sākumā noteikts skaits gredzenu nav savā starpā saistīti. Pēc tam gredzeni tiek savstarpēji savienoti. Jebkurā laika brīdī jums var palūgt noteikt kritisko gredzenu skaitu izveidotajā konfigurācijā. Precīzāk, jums jāizveido trīs procedūras.

- `Init`(N) — tiek izsaukta tieši vienreiz komunikācijas sākumā, lai paziņotu, ka sākotnējā konfigurācijā ir N savstarpēji nesaistīti gredzeni, kas sanumurēti ar veseiem skaitļiem no 0 līdz N - 1 (ieskaitot) pēc kārtas.
- `Link`(A, B) — divi gredzeni ar numuriem A un B tiek savienoti. Tiek garantēts, ka A un B ir atšķirīgi un pirms procedūras izpildes nav tieši savienoti. Bez tam, gredzenu A un B savienošanu neierobežo citi (tostarp fizikāli) ierobežojumi. Izsaukumi `Link`(A, B) un `Link`(B, A) ir ekvivalenti.

`CountCritical`() — atgriež kritisko gredzenu skaitu konfigurācijā.

Piemērs

Aplūkosim iepriekš doto zīmējumu ar N = 7 gredzeniem un pieņemsim, ka tie sākotnēji ir nesaistīti. Parādīsim iespējamo izsaukumu secību, kur pēc pēdējā izsaukuma tiks iegūta zīmējumā parādītā situācija.

Izsaukums	Atgriež
`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

1.apakšuzdevums [20 punkti]

- $N \leq 5\,000$.

Funkcija `CountCritical` tiek izsaukta tikai vienreiz pēc visiem pārējiem izsaukumiem; funkcija `Link` tiek izsaukta ne vairāk kā 5 000 reizes.

2.apakšuzdevums [17 punkti]

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

Funkcija `CountCritical` tiek izsaukta tikai vienreiz pēc visiem pārējiem izsaukumiem; funkcija `Link` tiek izsaukta ne vairāk kā 1 000 000 reizes.

3.apakšuzdevums [18 punkti]

- $N \leq 20\,000$.

Funkcija `CountCritical` tiek izsaukta ne vairāk kā 100 reizes; funkcija `Link` tiek izsaukta ne vairāk kā 10 000 reizes.

4.apakšuzdevums [14 punkti]

- $N \leq 100\,000$.

Funkcijas `CountCritical` un `Link` kopumā tiek izsauktas ne vairāk kā 100 000 reizes.

5.apakšuzdevums [31 punkts]

- $N \leq 100\,000$.

Funkcijas `CountCritical` un `Link` kopumā tiek izsauktas ne vairāk kā 1 000 000 reizes.

Implementācijas detaļas

Jums jāiesūta viens fails ar nosaukumu `rings.c`, `rings.cpp` vai `rings.pas`. Failā jābūt realizētām augstāk aprakstītajām apakšprogrammām lietojot sekojošas signatūras.

C/C++ programmām

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

Pascal programmām

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

Apakšprogrammām jādarbojas iepriekš aprakstītajā veidā. Protams var implementēt arī citas apakšprogrammas. Iesūtītas programmas nedrīkst neko rakstīt / lasīt no standarta izvada / ievada vai arī izmantot kādu citu failu.

Paraugvērtētājs

Paraugvērtētājs ielasa ievaddatus šādā formātā:

- 1.rindā: N, L;
- katrā no nākamajām L rindām:
  - -1 izsaukt `CountCritical`;
  - A, B procedūras `Link` parametri.

Paraugvērtētājs izdrukās visus `CountCritical` rezultātus.