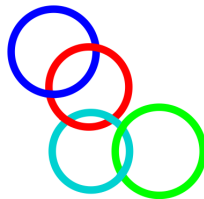


Anneaux de parachute (Parachute rings)

Une ancienne version relativement sophistiquée de ce que nous appelons maintenant un parachute est décrite dans le "Codex Atlanticus" (ca. 1485) de Léonard. Le parachute de Léonard consiste en un drap cousu sur une structure pyramidale en bois.

Anneaux connectés

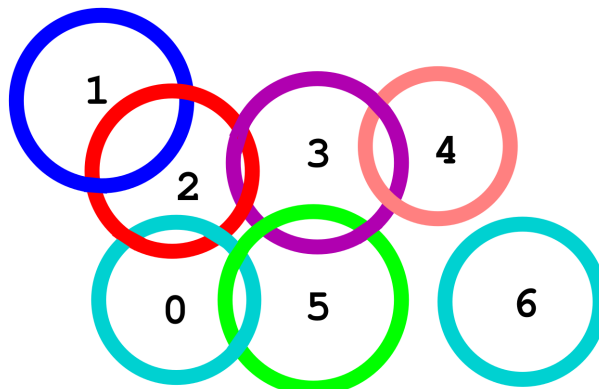
Plus de 500 ans plus tard, le parachutiste Nicholas Adrian a testé l'invention de Léonard. Il a relié le parachute de Léonard au corps humain par une structure moderne légère. Pour reproduire cette expérience, nous souhaitons utiliser un ensemble d'anneaux connectés les uns aux autres, et attachés au parachute par des crochets. Chaque anneau est fait d'un matériau flexible et robuste. Les anneaux peuvent facilement être connectés les uns aux autres car chaque anneau peut être ouvert puis refermé. Une configuration particulière d'anneaux connectés est la *chaîne*. Une chaîne est une séquence d'anneaux dans laquelle chaque anneau n'est connecté qu'à ses voisins (au plus deux), comme illustré ci-dessous. Cette séquence doit avoir un début et une fin (des anneaux qui sont connectés à un autre anneau au maximum). En particulier, un anneau seul est aussi une chaîne.



D'autres configurations sont certainement possibles, car un anneau peut être relié à trois ou plus autres anneaux. On dit qu'un anneau est "critique" si après l'avoir ouvert et retiré, tous les anneaux restants forment un ensemble de chaînes (ou qu'il ne reste plus aucun autre anneau). En d'autres mots, il ne peut rester que des chaînes.

Exemple

Considérez les 7 anneaux de l'illustration suivante, numérotés de 0 à 6. Il y a deux anneaux critiques. Un premier anneau critique est le 2 : après son retrait, les anneaux restants forment les chaînes [1], [0, 5, 3, 4], et [6]. L'autre anneau critique est 3 : après son retrait, les anneaux restants forment les chaînes [1, 2, 0, 5], [4] et [6]. Si l'on retire tout autre anneau, on n'obtient pas un ensemble de chaînes disjointes. Par exemple, lorsque l'on retire l'anneau 5, on a bien une chaîne [6], mais l'ensemble d'anneaux connectés 0, 1, 2, 3 et 4 ne forme pas une chaîne.



Problème

Votre objectif est de compter le nombre d'anneaux critiques dans une configuration donnée qui sera communiquée à votre programme.

Au début, il y a un certain nombre d'anneaux disjoints. Après cela, les anneaux seront connectés les uns aux autres. À tout moment, on peut vous demander de retourner le nombre d'anneaux critiques de la configuration actuelle. Plus précisément, vous devez implémenter trois fonctions :

- `Init(N)` - est appelé exactement une fois au début pour indiquer qu'il y a N anneaux disjoints numérotés de 0 à N-1 (inclus) dans la configuration initiale.
- `Link(A, B)` - les deux anneaux numérotés A et B sont connectés l'un à l'autre. On vous garantit que A et B sont différents et ne sont pas déjà reliés directement. À part cela, il n'y a pas de conditions supplémentaires sur A et B, et en particulier pas de conditions dues à des contraintes physiques. Bien sûr, `Link(A, B)` et `Link(B, A)` sont équivalents.
- `CountCritical()` - retourne le nombre d'anneaux critiques pour la configuration courante des anneaux connectés

Exemple

Considérez notre illustration avec N = 7 anneaux et supposez qu'ils ne sont pas connectés au départ. On vous montre une séquence possible d'appels, telle qu'après le dernier appel, on obtient la situation décrite sur notre illustration.

Appel	Retourne
<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

Sous-tâche 1 [20 points]

- N ≤ 5 000.
- La fonction `CountCritical` n'est appelée qu'une seule fois, après tous les autres appels ; la fonction `Link` est appelée au maximum 5000 fois.

Sous-tâche 2 [17 points]

- N ≤ 1 000 000.
- La fonction `CountCritical` n'est appelée qu'une seule fois, après tous les autres appels ; la fonction `Link` est appelée au maximum 1 000 000 fois.

Sous-tâche 3 [18 points]

- $N \leq 20\,000$
- La fonction `CountCritical` est appelée au maximum 100 fois ; la fonction `Link` est appelée au maximum 10 000 fois.

Sous-tâche 4 [14 points]

- $N \leq 100\,000$
- Les fonctions `CountCritical` et `Link` sont appelées au maximum 100 000 fois au total.

Sous-tâche 5 [31 points]

- $N \leq 1\,000\,000$
- Les fonctions `CountCritical` et `Link` sont appelées au maximum 1 000 000 fois au total.

Détails d'implémentation

Vous devez soumettre exactement un fichier, appelé `rings.c`, `rings.cpp` ou `rings.pas`. Ce fichier implémente les fonction décrites ci-dessus en utilisant les signatures qui suivent.

Programmes C/C++

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

Programmes Pascal

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

Ces fonctions doivent se comporter comme décrit plus haut. Bien sûr vous êtes libre d'implémenter d'autres fonctions pour un usage interne. Vos soumissions ne doivent en aucun cas interagir avec les entrées/sorties ni avec tout autre fichier.

Évaluateur d'exemple

L'évaluateur d'exemple lit l'entrée au format suivant :

- ligne 1 : N, L ;
- lignes 2, ..., $L + 1$:
 - -1 pour appeler `CountCritical`
 - A, B : les paramètres de `Link`

L'évaluateur d'exemple affichera tous les résultats des appels à `CountCritical`.