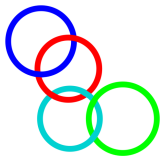
Anillos de paracaidas

Una versión más antigua y menos sofisticada de lo que ahora llamamos paracaídas fue descrita en el “Codex Atlantius” (ca. 1485) de Leonardo. El paracaídas de Leonardo consistía en una tela de lino sellada, abierta por una estructura de madera en forma de pirámide.

Anillos enlazados

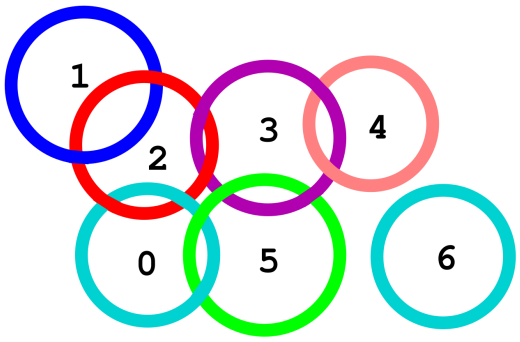
El paracaidista Adrian Nicholas probó el diseño de Leonardo 500 años después. Para esto, usó una estructura moderna y ligera del paracaidas de Leonardo ajustada al cuerpo humano. Para la estructura queremos usar anillos enlazados, que también proveerán ganchos para la tela de lino sellada. Cada anillo está hecho de un material fuerte y flexible. Los anillos pueden ser fácilmente enlazados de tal manera que cada anillo se puede abrir y volver a cerrar. Una configuración especial de anillos enlazados es llamada “cadena”: una “cadena” es una secuencia de uno o más anillos enlazados donde cada anillo está enlazado a otros dos anillos, excepto el último y el primer anillo que están enlazados solo a otro anillo, como se muestra en la imagen inferior. La secuencia de anillos debe tener un inicio y un final (el inicio y final son anillos conectados a lo más a otro anillo). Particularmente un solo anillo es una "cadena".



Otras configuraciones son claramente posibles pues un anillo puede ser enlazado a tres o más anillos. Decimos que un anillo es “crìtico” si después de abrirlo y removerlo, los anillos restantes forman un conjunto disjunto de “cadenas” (o si no queda algún otro anillo). En otras palabras, después de remover el anillo solo pueden quedar cadenas.

Ejemplo

Considera los 7 anillos de la siguiente imagen, numerados de 0 a 6. En esa imagen existen dos anillos “críticos”. Un anillo “crítico” es el número 2: después de removerlo, los anillos sobrantes forman las “cadenas” [1], [0, 5, 3, 4] y [6]. El otro anillo “crítico” es el numero 3: después de removerlo, los anillos sobrantes forman las “cadenas” [1, 2, 0, 5], [4] y [6]. Si removemos cualquier otro anillo, no obtenemos un conjunto de “cadenas” disjuntas. Por ejemplo, después de remover el anillo 5: aunque tenemos que [6] es una “cadena”, los anillos enlazados 0,1,2,3 y 4 no forman una “cadena”.



Problema

Tu tarea es contar el número de anillos “críticos” de una configuración dada, que será comunicada a tu programa.

Al inicio, existen cierto número de anillos disjuntos. Después de eso, los anillos son enlazados entre sí. En cualquier momento, se te puede preguntar el número de anillos “críticos” de la configuración actual. Específicamente, tienes que implementar tres funciones:

- `Init`(N) — Es llamada exactamente una vez al inicio para comunicar que existen N anillos disjuntos numerados de 0 a N-1 (inclusive) en la configuración inicial.
- `Link`(A, B) —los dos anillos numerados A y B son enlazados entre sí. Se garantiza que A y B son diferentes y no se encuentran actualmente enlazados; aparte de esto, no existe ninguna otra condición para enlazar en A y B, en particular no existe ningún impedimento físico para enlazar los anillos. Claramente Link(A, B) y Link(B, A) son equivalentes.
- `CountCritical`() — regresa el número de anillos “críticos” de la configuración actual de anillos enlazados.

Ejemplo

Considera la imagen anterior con N=7 anillos y supón que inicialmente ningún anillo está enlazado. Mostramos una posible secuencia de llamadas, donde después de la última llamada obtenemos la situación descrita en la imagen.

Llamada	Devuelve
`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

Sub-tarea 1 [20 puntos]

- $N \leq 5\,000$.
- La funcion `CountCritical`es llamada sola una vez después de todas las demás llamadas; la función `Link` es llamada a lo más 5 000 veces.

Sub-tarea 2 [17 puntos]

- $N \leq 1\,000\,000$.
- La funcion `CountCritical` es llamada una sola vez después de todas las demás llamadas; la función `Link` es llamada a lo más 1 000 000 veces.

Sub-tarea 3 [18 puntos]

- $N \leq 20\,000$.
- La funcion `CountCritical` es llamada a lo más 100 veces; la función `Link` es llamada a lo más 10 000 veces.

Sub-tarea 4 [14 puntos]

- $N \leq 100\,000$.
- La funcion `CountCritical` y `Link` son llamadas, en total, a lo sumo 100 000 veces.

Sub-tarea 5 [31 puntos]

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

- La funcion `CountCritical` y `Link` son llamadas, en total, a lo sumo 1 000 000 veces.

Detalles de implementación

Tienes que mandar exactamente un archivo llamado `rings.c`, `rings.cpp` or `rings.pas`. El archivo tiene implementadas las funciones descritas arriba usando los siguientes encabezados:

Programas en C/C++

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

Programas en Pascal

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

Estas funciones deben tener el comportamiento descrito arriba. Puedes implementar otras funciones para uso interno. Tus envíos no pueden interactuar de ninguna manera con la salida y entrada estándar o algún otro archivo.

Ejemplo de evaluador (Sample grader)

El evaluador de ejemplo lee la entrada con el siguiente formato:

- linea 1: N, L;
- lineas 2, …, L + 1:
  - -1 para llegar `CountCritical`;
  - A, B parametros to `Link`.

El evaluador de ejemplo imprimirá los resultados de `CountCritical`.