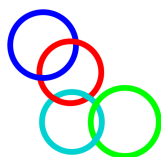


# Kruhové padáky

Stará a celkom rozumná verzia toho, čo dnes nazývame padák, je opísaná v Leonardovom spise *Codex Atlanticus* (ca. 1485). Leonardov padák pozostával z plachtoviny, ktorá bola otváraná a spevnená pomocou drevenej štruktúry pyramidového tvaru.

## Spojené kruhy

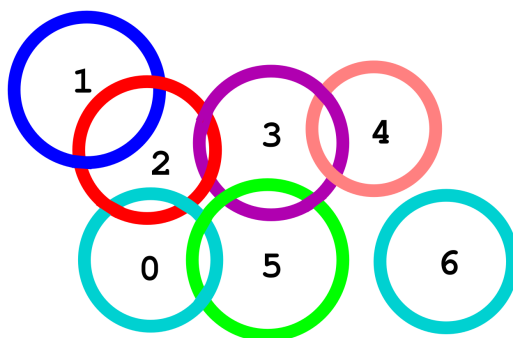
Parašutista Adrian Nicholas testoval Leonardov návrh o viac než 500 rokov neskôr. Leonardov padák mal k telu pripojený modernou, málo vážiacou štruktúrou reťazí. Krúžky týchto reťazí boli urobené z flexibilného a silného materiálu. Kruhy je možné ľahko spájať, pretože každý kruh možno otvoriť a znovu zatvoriť. Špeciálna konfigurácia spojených kruhov je *reťaz*. Reťaz je postupnosť kruhov, v ktorej je každý kruh spojený len so svojimi (najviac dvoma) susedmi tak, ako je to znázornené nižšie. Táto postupnosť musí mať začiatok a koniec (začiatočný a tiež koncový kruh je spojený s najviac jedným iným kruhom). Špeciálne, jeden izolovaný kruh je tiež reťaz.



Okrem reťazí samozrejme existujú aj iné konfigurácie, pretože každý kruh môže byť spojený aj s tromi a viac inými kruhmi. Hovoríme, že kruh je *kritický*, ak po jeho otvorení a odstránení všetky zostávajúce kruhy tvoria množinu reťazí. (To nastane aj vtedy, ak už nezostalo vôbec nič.) Inak povedané, nesmie zostať žiadna po kope držiaca konfigurácia kruhov, ktorá nie je reťaz.

## Príklad

Na nasledujúcom obrázku je 7 kruhov očíslovaných od 0 do 6. V tejto štruktúre existujú dva kritické kruhy. Jedným z nich je kruh 2: po jeho odstránení zostávajúce kruhy tvoria reťaze [1], [0, 5, 3, 4] a [6]. Druhý kritický kruh je 3: po jeho odstránení zostávajúce kruhy tvoria reťaze [1, 2, 0, 5], [4] a [6]. Ak odstránime ľubovoľný iný kruh, nedostaneme množinu disjunktných reťazí. Napríklad, po odstránení kruhu 5: hoci [6] je reťaz, spojené kruhy 0, 1, 2, 3 a 4 netvoria reťaz.



## Úloha

Vašou úlohou je vypočítať počet kritických kruhov v danej konfigurácii, ktorá bude analyzovaná vaším programom.

Na začiatku je daný určitý počet kruhov, pričom žiadne dva nie sú prepojené. Potom kruhy postupne spájame. V ľubovoľnom okamihu môže byť od vás požadovaná odpoveď na otázku o počte kritických kruhov v aktuálnej konfigurácii. Špecificky, musíte implementovať tri rutiny.

- `Init(N)` — je zavolaná práve jedenkrát na začiatku komunikácie, pričom  $N$  je počet disjunktných kruhov očíslovaných od 0 do  $N - 1$  (včítane) v počiatočnej konfigurácii.
- `Link(A, B)` — dva kruhy s číslami  $A$  a  $B$  sa stávajú spojenými. Je zaručené, že  $A$  a  $B$  sú rôzne a ešte nie sú priamo spojené.  $A$  a  $B$  nemusia spĺňať žiadne iné podmienky. Špeciálne nepredpokladajte nič vyplývajúce z fyzikálnych obmedzení. Volania `Link(A, B)` a `Link(B, A)` sú ekvivalentné.
- `CountCritical()` — vráti počet kritických kruhov pre aktuálnu konfiguráciu spojených kruhov.

## Príklad

Majme konfiguráciu tvorenú  $N = 7$  kruhmi a predpokladajme, že na začiatku nie sú spojené. Ukážeme možnú postupnosť volaní, pričom po poslednom volaní dostaneme situáciu znázornenú na našom obrázku.

Volanie	Návraty
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

## Podúloha 1 [20 bodov]

- $N \leq 5\,000$ .
- Funkcia `CountCritical` je zavolaná len jedenkrát, po všetkých iných volaniach; funkcia `Link` je zavolaná najviac 5 000 krát.

## Podúloha 2 [17 bodov]

- $N \leq 1\,000\,000$ .
- Funkcia `CountCritical` je zavolaná len jedenkrát, po všetkých ostatných volaniach; funkcia `Link` je zavolaná najviac 1 000 000 krát.

## Podúloha 3 [18 bodov]

- $N \leq 20\,000$ .
- Funkcia `CountCritical` je zavolaná najviac 100 krát; funkcia `Link` je zavolaná najviac 10 000 krát.

## Podúloha 4 [14 bodov]

- $N \leq 100\,000$ .
- Funkcie `CountCritical` a `Link` sú zavolané celkom, najviac 100 000 krát.

## Podúloha 5 [31 bodov]

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

- Funkcie `CountCritical` a `Link` sú zavolané celkom, najviac 1 000 000 krát.

## Implementačné detaily

Musíte poslať do systému práve jeden súbor, ktorý sa volá `rings.c`, `rings.cpp` alebo `rings.pas`. Tento súbor musí implementovať podprogramy, ktoré sú popísané vyššie použitím nasledujúcich označení.

### C/C++ programy

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

### Pascal programy

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

Tieto podprogramy sa musia správať tak, ako je uvedené vyššie. Samozrejme, môžete implementovať iné podprogramy pre svoju internú potrebu. Váš odoslaný súbor nesmie komunikovať žiadnym spôsobom so štandardným vstupom/výstupom a ani s inými súbormi.

### Hodnotenie príkladu

Pri vyhodnotení príkladu je prečítaný vstup v nasledujúcom formáte:

- riadok 1:  $N, L$ ;
- riadky 2, ...,  $L + 1$ :
  - -1 spustiť `CountCritical`;
  - $A, B$  parametre do `Link`.

Ukážkový testovač vypíše na výstup všetky výsledky z `CountCritical`.