



Механичка кукла

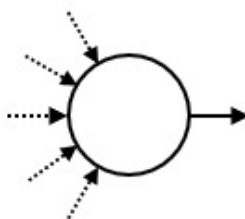
Механичка кукла е кукла која автоматски повторува определена секвенца од движења. Во Јапонија, уште од антички времиња, направени се многу механички кукли.

Движењата на механичката кукла се контролирани од страна на **коло** (анг. **circuit**) кое што се состои од **уреди** (анг. **devices**). Уредите се поврзани со цевки. Секој уред има еден или два **излези**, и може да има произволен број (може да е и 0) **влезови**. Секоја цевка поврзува излез на даден уред со влезот на истиот или на некој друг уред. Со секој влез и со секој излез е поврзана точно една цевка.

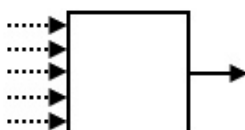
За да опишеме како куклата прави движења, замислете дека една **топка** е поставена на еден од уредите. Топката патува низ колото. Во секој чекор од патувањето, топката го напушта уредот користејќи еден од неговите излези, патува долж цевката поврзана со излезот и влегува во уредот на другиот крај од цевката.

Постојат три типа на уреди: **извор** (анг. **origin**), **поттикнувач** (анг. **trigger**) и **префрлувач** (анг. **switch**). Постојат точно еден извор, M поттикнувачи и S префрлувачи (S може да биде и 0). Треба да ја определите вредноста на S . Секој уред има единствен (уникатен) сериски број.

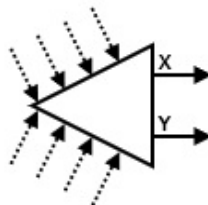
Извор е уредот во кој се поставува топката на почетокот. Тој има само еден излез. Неговиот сериски број е 0.



Поттикнувачот предизвикува куклата да направи некакво движење секогаш кога топката ќе влезе во него. Секој поттикнувач има еден излез. Сериските броеви на поттикнувачите се од 1 до M .



Секој префрлувач има два излеза, кои се именувани 'X' и 'Y'. **Состојбата** на даден префрлувач е или 'X' или 'Y'. Откако топката ќе влезе во префрлувачот, таа го напушта префрлувачот користејќи го излезот даден со тековната состојба на префрлувачот. Потоа, префрлувачот ја менува својата состојба во спротивната. На почетокот, состојбата на секој префрлувач е 'X'. Сериските броеви на префрлувачите се од -1 до $-S$.



Даден ви е бројот на поттикнувачи M . Исто така, дадена ви е и низа A со должина N , чии што елементи се сериски броеви на поттикнувачи. Секој поттикнувач може да се појави неколку (може да биде и 0) пати во низата A . Ваша задача е да креирате коло кое што ќе ги задоволува следните услови:

- Топката се враќа во изворот по одреден број на чекори.
- Кога топката за прв пат ќе се врати во изворот, состојбата на секој префрлувач е 'X'.
- Топката за прв пат се враќа во изворот откако ќе влезе во поттикнувачи точно N пати. Сериските броеви на поттикнувачите, во редоследот по кој топката влегува во нив, се A_0, A_1, \dots, A_{N-1} .
- Нека P е вкупниот број на промени на состојба на сите префрлувачи предизвикани од страна на топката пред таа за прв пат да се врати во изворот. Вредноста на P не надминува 20 000 000.

Во исто време, не треба да користите премногу префрлувачи.

Имплементациски детали

Треба да ја имплементирате следнава процедура:

```
create_circuit(int M, int[] A)
```

- M : бројот на поттикнувачи.
- A : низа со должина N , која што ги претставува сериските броеви на поттикнувачите, по редослед по кој топката треба да влезе во нив.
- Оваа процедура се повикува точно еднаш.
- Да забележиме дека вредноста на N е должината на низата A и истата може да се добие како што е објаснето во Забелешки за имплементацијата (Ден 2).

Вашата програма треба да ја повика следнава процедура за да даде одговор:

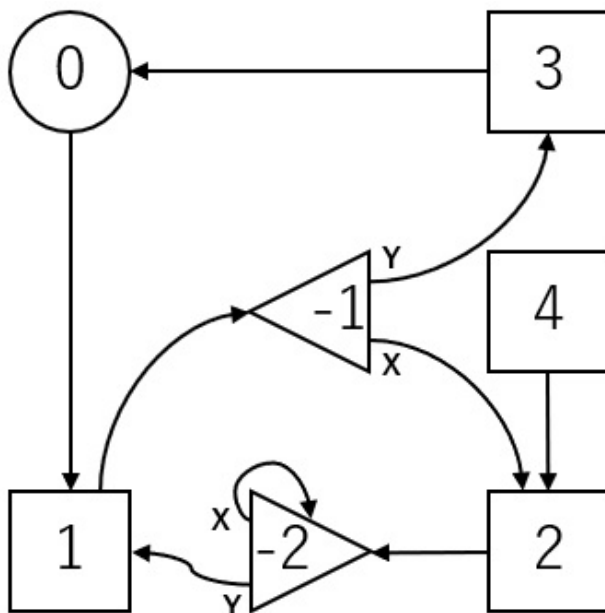
```
answer(int[] C, int[] X, int[] Y)
```

- C : низа со должина $M + 1$. Излезот на уредот i ($0 \leq i \leq M$) е поврзан со уредот $C[i]$.
- X, Y : низи со иста должина. Должината S на овие низи е бројот на префрлувачи. За префрлувачот $-j$ ($1 \leq j \leq S$) важи дека неговиот излез 'X' е поврзан со уредот $X[j - 1]$, додека неговиот излез 'Y' е поврзан со уредот $Y[j - 1]$.
- Секој од елементите на C, X и Y мора да биде цел број помеѓу $-S$ и M , вклучително.
- S мора да биде најмногу 400 000.
- Оваа процедура мора да се повика точно еднаш.
- Колото претставено со C, X и Y мора да ги задоволува условите од описот на задачата.

Ако некои од дадените услови не се задоволени, вашата програма се оценува како **Wrong Answer**. Во спротивно, вашата програма се оценува како **Accepted** и вашиот резултат се пресметува според S (видете го делот Подзадачи).

Пример

Нека $M = 4, N = 4$ и $A = [1, 2, 1, 3]$. Оценувачот ја повикува процедурата `create_circuit(4, [1, 2, 1, 3])`.



На сликата погоре е прикажано коло кое што е опишано со повикот `answer([1, -1, -2, 0, 2], [2, -2], [3, 1])`. Броевите на сликата се сериските броеви на уредите.

Се користат два префрлувачи. Според тоа, $S = 2$.

На почеток, состојбата и на двата префрлувачи -1 и -2 е 'X'.

Топката пагува вака:

$$0 \longrightarrow 1 \xrightarrow{X} -1 \xrightarrow{X} 2 \xrightarrow{X} -2 \xrightarrow{Y} -2 \xrightarrow{Y} 1 \xrightarrow{Y} -1 \xrightarrow{Y} 3 \longrightarrow 0$$

- Кога топката за прв пат ќе влезе во префрлувачот -1 , неговата состојба е 'X'. Според тоа, топката оди кон поттикнувачот 2 . Потоа, состојбата на префрлувачот -1 се менува во 'Y'.
- Кога топката по втор пат ќе влезе во префрлувачот -1 , неговата состојба е 'Y'. Според тоа, топката оди кон поттикнувачот 3 . Потоа, состојбата на префрлувачот -1 се менува во 'X'.

Топката за прв пат се враќа во изворот откако влегла во поттикнувачите $1, 2, 1, 3$. Состојбата и на двата префрлувачи -1 и -2 е 'X'. Вредноста на P е 4 . Според тоа, ова коло ги задоволува бараните услови.

Датотеката `sample-01-in.txt` во zip архивата одговара на овој пример. Во архивата исто така се достапни и други пример-влезови.

Ограничувања

- $1 \leq M \leq 100\,000$
- $1 \leq N \leq 200\,000$
- $1 \leq A_k \leq M$ ($0 \leq k \leq N - 1$)

Подзадачи

Резултатот и ограничувањата за секој од тест случаите се следниве:

1. (2 поени) За секое i ($1 \leq i \leq M$), целиот број i се појавува најмногу еднаш во низата A_0, A_1, \dots, A_{N-1} .
2. (4 поени) За секое i ($1 \leq i \leq M$), целиот број i се појавува најмногу два пати во низата A_0, A_1, \dots, A_{N-1} .
3. (10 поени) За секое i ($1 \leq i \leq M$), целиот број i се појавува најмногу 4 пати во низата A_0, A_1, \dots, A_{N-1} .
4. (10 поени) $N = 16$
5. (18 поени) $M = 1$
6. (56 поени) Нема дополнителни ограничувања

За секој тест случај, ако вашата програма е оценета како **Accepted**, вашиот резултат се пресметува според вредноста на S :

- Ако $S \leq N + \log_2 N$, ги добивате сите поени за тест случајот.
- За секој тест случај во Подзадачите 5 и 6, ако $N + \log_2 N < S \leq 2N$, добивате

делумни поени. Резултатот за тест случајот е $0.5 + 0.4 \times \left(\frac{2N - S}{N - \log_2 N} \right)^2$, помножено со бројот на поени што ги носи подзадачата.

- Во секој друг случај, резултатот е 0.

Да забележиме дека вашиот резултат за секоја подзадача е минимумот од резултатите за тест случаите во таа подзадача.

Пример оценувач

Пример оценувачот го чита влезот од стандардниот влез во следниот формат:

- линија 1: $M N$
- линија 2: $A_0 A_1 \dots A_{N-1}$

Пример оценувачот продуцира три излези.

Прво, пример оценувачот го печати вашиот одговор во датотека со име `out.txt` во следниот формат:

- линија 1: S
- линии $2 + i$ ($0 \leq i \leq M$): $C[i]$
- линии $2 + M + j$ ($1 \leq j \leq S$): $X[j - 1] Y[j - 1]$

Второ, пример оценувачот ги симулира движењата на топката. Тој ги печати сериските броеви на уредите по редоследот на влегување на топката, во датотека со име `log.txt`.

Трето, пример оценувачот ја печати евалуацијата на вашиот одговор на стандардниот излез.

- Ако вашата програма е оценета како **Accepted**, пример оценувачот ги печати S и P во следниот формат `Accepted: S P`.
- Ако вашата програма е оценета како **Wrong Answer**, пример оценувачот печати `Wrong Answer: MSG`. Значењето на `MSG` е следново:
 - `answered not exactly once`: Процедурата `answer` не е повикана точно еднаш.
 - `wrong array length`: Должината на C не е $M + 1$, или пак должините на X и Y се различни.
 - `over 400000 switches`: S е поголемо од 400 000.
 - `wrong serial number`: Постои елемент на C , X или Y кој што е помал од $-S$ или поголем од M .
 - `over 20000000 inversions`: Топката не се враќа во изворот во рамки на 20 000 000 промени на состојба на префрлувачите.
 - `state 'Y'`: Постои префрлувач чија што состојба е 'Y' во моментот кога

топката за прв пат се враќа во изворот.

- `wrong motion`: Поттикнувачите кои предизвикуваат движења се разликуваат во однос на низата A .

Да забележиме дека пример оценувачот може да не ја креира датотеката `out.txt` и/или `log.txt` во случај кога вашата програма е оценета како `Wrong Answer`.