



Mehanična lutka

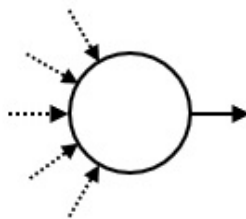
Mehanična lutka je lutka, ki samodejno ponovi določeno zaporedje gibov. Na Japonskem so od davnine do današnjih dni izdelali že veliko mehaničnih lutk.

Gibe mehanične lutke krmili **vezje**, sestavljeno iz **naprav**. Naprave so povezane s cevmi. Vsaka naprava ima eden ali dva **izhoda** in poljubno mnogo (lahko tudi 0) **vhodov**. Vsaka cev povezuje izhod neke naprave z vhomom iste ali neke druge naprave. Na vsak vhod je povezana natanko ena cev, prav tako pa je na vsak izhod povezana natanko ena cev.

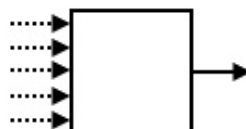
Za opis, kako lutka izvaja gibe, si predstavljaj **žogo**, ki potuje po vezju med napravami. Na vsakem koraku potovanja žoga izstopi iz naprave na enem izmed njenih izhodov, potuje po cevi in vstopi v napravo na drugem koncu cevi.

Naprave so treh vrst: **izhodišče**, **sprožilec** in **stikalo**. Vezje je sestavljeno iz natanko enega izhodišča, M sprožilcev in S stikal (vrednost S je lahko tudi 0). Vrednost S določi sam(-a). Vsaka naprava ima enolično zaporedno številko.

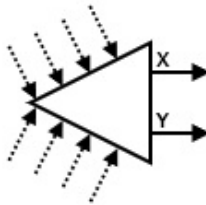
Izhodišče je naprava, v kateri se na začetku nahaja žoga. Ta naprava ima en izhod, njena zaporedna številka pa je 0.



Sprožilec povzroči, da lutka izvede določen gib vsakokrat, ko vanj vstopi žoga. Vsak sprožilec ima natanko en izhod. Zaporedne številke sprožilcev so od 1 do M .



Vsako stikalo ima dva izhoda; imenujeta se 'X' in 'Y'. **Stanje** stikala je bodisi 'X' bodisi 'Y'. Potem ko žoga vstopi v stikalo, izstopi iz njega na izhodu, ki ga določa trenutno stanje stikala. Takoj zatem stikalo spremeni svoje stanje v nasprotno stanje. Na začetku je vsako stikalo v stanju 'X'. Zaporedne številke stikal so od -1 do $-S$.



Podano je število sprožilcev M in zaporedje A dolžine N . Vsak člen zaporedja je zaporedna številka nekega sprožilca. Vsak sprožilec lahko v zaporedju A nastopa poljubno mnogokrat (lahko tudi 0-krat). Tvoja naloga je izdelati vezje, ki izpolnjuje naslednje pogoje:

- Žoga se po določenem številu korakov vrne v izhodišče.
- Ko se žoga prvič vrne v izhodišče, so vsa stikala v stanju 'X'.
- Žoga se prvič vrne v izhodišče po natanko N vstopih v sprožilce. Zaporedne številke sprožilcev, v katere žoga vstopi, so A_0, A_1, \dots, A_{N-1} (v vrstnem redu vstopov v sprožilce).
- Naj bo P skupno število sprememb stanj vseh stikal, ki jih povzroči žoga, preden se prvič vrne v izhodišče. Vrednost P ne presega 20 000 000.

Poleg naštetega želiš vezje s čim manj stikali.

Podrobnosti implementacije

Implementiraj naslednjo proceduro:

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

- M : število sprožilcev.
- A : polje dolžine N , ki podaja zaporedje zaporednih številok sprožilcev, v katere mora vstopiti žoga (v vrstnem redu vstopov v sprožilce).
- Ta procedura se pokliče natanko enkrat.
- Upoštevaj, da je vrednost N enaka dolžini polja A in da jo lahko pridobiš, kot je navedeno v implementacijskih opombah.

Tvoj program mora sporočiti rešitev naloge tako, da pokliče naslednjo proceduro:

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

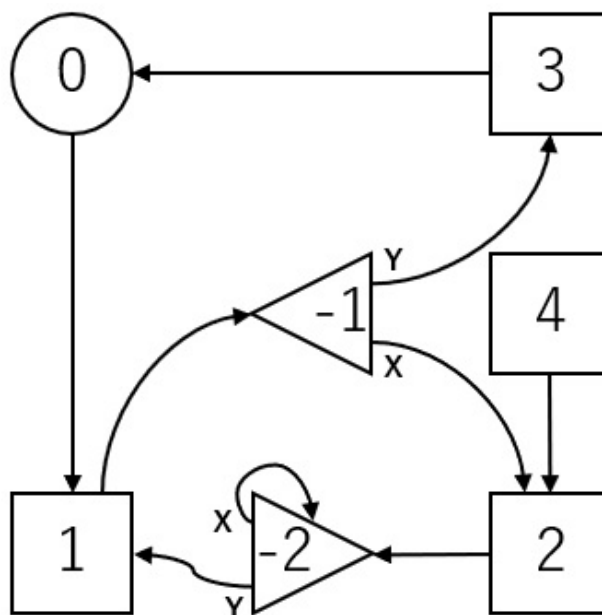
- C : polje dolžine $M + 1$. Izhod naprave i ($0 \leq i \leq M$) je povezan na vhod naprave $C[i]$.
- X, Y : polji iste dolžine. Dolžina S teh polj je enaka številu stikal. Izhod 'X' stikala $-j$ ($1 \leq j \leq S$) je povezan na napravo $X[j - 1]$, njegov izhod 'Y' pa na napravo $Y[j - 1]$.
- Vsak element polj C, X in Y mora biti celo število med $-S$ in vključno M .

- Vrednost S mora biti kvečjemu 400 000.
- To proceduro moraš poklicati natanko enkrat.
- Vezje, ki ga določajo polja C , X in Y , mora izpolnjevati pogoje, navedene v besedilu naloge.

Če kateri od zgornjih pogojev ni izpolnjen, se ocenjevalnik odzove z **Wrong Answer**. V nasprotnem primeru je odziv **Accepted**, število točk pa se izračuna na podlagi vrednosti S (glej razdelek Podnaloge).

Primer

Naj bo $M = 4$, $N = 4$ in $A = [1, 2, 1, 3]$. Ocenjevalnik pokliče `create_circuit(4, [1, 2, 1, 3])`.



Zgornja slika prikazuje vezje, opisano s klicem `answer([1, -1, -2, 0, 2], [2, -2], [3, 1])`. Številke na sliki so zaporedne številke naprav.

Uporabljamo dve stikali. Zato je $S = 2$.

Na začetku sta stikali -1 in -2 obe v stanju 'X'.

Žoga potuje na naslednji način:

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

- Ko žoga prvič vstopi v stikalo -1 , je njegovo stanje enako 'X'. Zato žoga potuje v sprožilec 2. Stanje stikala -1 se spremeni v 'Y'.
- Ko žoga drugič vstopi v stikalo -1 , je njegovo stanje enako 'Y'. Zato žoga potuje v sprožilec 3. Stanje stikala -1 se spremeni v 'X'.

Žoga se prvič vrne v izhodišče, potem ko je vstopila v sprožilce 1, 2, 1, 3. Stikali -1 in

–2 sta obe v stanju 'X'. Vrednost P znaša 4. Vežje potemtakem izpolnjuje pogoje.

Na pravkar prikazani primer se nanaša datoteka `sample-01-in.txt` v priloženem paketu zip. Tudi ostali vzorčni vhodi so na voljo v paketu.

Omejitve

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

Podnaloge

Točke in omejitve so za vsak testni primer podane takole:

1. (2 točki) Za vsak i ($1 \leq i \leq M$) se celo število i v zaporedju A_0, A_1, \dots, A_{N-1} pojavi največ enkrat.
2. (4 točke) Za vsak i ($1 \leq i \leq M$) se celo število i v zaporedju A_0, A_1, \dots, A_{N-1} pojavi največ dvakrat.
3. (10 točk) Za vsak i ($1 \leq i \leq M$) se celo število i v zaporedju A_0, A_1, \dots, A_{N-1} pojavi največ 4-krat.
4. (10 točk) $N = 16$.
5. (18 točk) $M = 1$.
6. (56 točk) Ni dodatnih omejitev.

Za vsak testni primer velja: če se odziv na tvoj program glasi **Accepted**, se dobljene točke izračunajo na podlagi vrednosti S :

- Če velja $S \leq N + \log_2 N$, potem za ta testni primer prejmeš vse točke.
- Za vsak testni primer v podnalogah 5 in 6 velja: če je $N + \log_2 N < S \leq 2N$, prejmeš nepopolno število točk. Število točk za testni primer je enako $0.5 + 0.4 \times \left(\frac{2N - S}{N - \log_2 N} \right)^2$, pomnoženo s številom točk, dodeljenih podnalogi.
- Če ne velja nič od naštetega, prejmeš 0 točk.

Upoštevaj, da se število točk za vsako podnalogo izračuna kot minimum števila točk za posamezne testne primere v podnalogi.

Vzorčni ocenjevalnik

Vzorčni ocenjevalnik bere vhod s standardnega vhoda v naslednji obliki:

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

Vzorčni ocenjevalnik proizvede tri izhode.

Prvič: vzorčni ocenjevalnik izpiše tvoj odgovor v datoteko z imenom `out.txt` v naslednji obliki:

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

Drugič: vzorčni ocenjevalnik simulira potovanje žoge. V datoteko z imenom `log.txt` po vrsti izpiše zaporedne številke naprav, v katere žoga vstopi.

Tretjič: vzorčni ocenjevalnik izpiše oceno tvojega odgovora na standardni izhod.

- Če je odziv na tvoj program **Accepted**, potem vzorčni ocenjevalnik izpiše vrednosti S in P v naslednji obliki `Accepted: S P`.
- Če je odziv na tvoj program **Wrong Answer**, potem izpiše `Wrong Answer: MSG`. Niz `MSG` pomeni:
 - `answered not exactly once`: Procedura `answer` se ne pokliče natanko enkrat.
 - `wrong array length`: Dolžina polja C ni enaka $M + 1$, ali pa se dolžini polj X in Y med seboj razlikujeta.
 - `over 400000 switches`: Vrednost S je večja od 400 000.
 - `wrong serial number`: Vsaj en element polja C , X ali Y je manjši od $-S$ ali večji od M .
 - `over 20000000 inversions`: Žoga se ne vrne v izhodišče niti po 20 000 000 spremembah stanj stikal.
 - `state 'Y'`: Ko se žoga prvič vrne v izhodišče, je vsaj eno stikalo v stanju 'Y'.
 - `wrong motion`: Sprožilci, ki povzročajo gibe, so različni od zaporedja A .

Ko je odziv na tvoj program `Wrong Answer`, ni nujno, da vzorčni ocenjevalnik ustvari datoteko `out.txt` in/ali `log.txt`.