



# Mechanical Doll

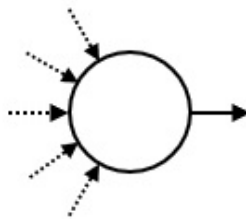
Lutke koje mogu automatizovano ponavljati specifični niz poteza nazivamo mehaničkim lutkama. U Japanu su još od drevnih vremena konstruisane drevne lutke od strane drevnog Šebeza.

Kretanje mehaničke lutke kontroliše **mehanizam** koje se sastoji od **uređaja (komponenti)**. Uređaji su međusobno povezani cevima. Svaki uređaj ima nula ili više **ulaza**, i jedan ili dva **izlaza**. Svaki uređaj može imati proizvoljno mnogo ulaza. Svaka cev povezuje izlaz jednog uređaja sa ulazom tog istog ili nekog drugog uređaja. Na svaki ulaz je povezana tačno jedna cev i na svaki izlaz je povezana tačno jedna cev.

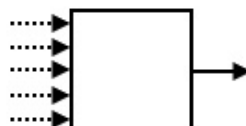
Da bi opisali način na koji se lutka pokreće, razmotrimo **kuglicu** koja se postavlja u neki od uređaja mehanizma. Kuglica sada putuje kroz mehanizam. U svakom koraku putovanja, loptica napušta uređaj kroz neki od izlaza, kreće se kroz cev koja je povezana na taj izlaz i ulazi u uređaj koji je na drugom kraju cevi.

Postoje tri tipa uređaja: **izvor (origin)**, **okidač (trigger)** i **razdelnik (switch)**. U mehanizmu postoji tačno jedan izvor,  $M$  okidača i  $S$  razdelnika ( $S$  može biti i nula). Vi morate da odredite kolika će biti vrednost  $S$ . Svaki uređaj ima jedinstveni serijski broj.

Izvor je uređaj u kome se kuglica postavlja na početku. On ima samo jedan izlaz i njegov serijski broj je 0.

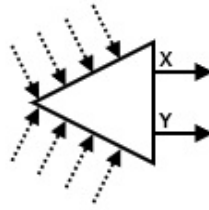


Okidač je uređaj koji omogućava da lutka izvrši jedan tip pokreta kada god kuglica uđe u njega. Svaki okidač ima tačno jedan izlaz. Serijski brojevi okidača su od 1 do  $M$ .



Svaki razdelnik ima dva izlaza, koje označavamo sa 'X' i 'Y'. **Stanje** razdelnika može biti ili 'X' ili 'Y'. Kada kuglica uđe u razdelnik, ona izlazi iz njega na onaj izlaz koji je označen trenutnim stanjem razdelnika. Po izlasku kuglice, razdelnik menja stanje u

suprotno. Na početku, svaki razdelnik se nalazi u stanju 'X'. Serijski brojevi razdelnika su negativni, od  $-1$  do  $-S$ .



Dat vam je broj okidača  $M$  i niz  $A$  dužine  $N$  čiji su elementi serijski brojevi okidača. Svaki okidač se može pojaviti u nizu  $A$  nula ili više puta. Vaš zadatak je da pomognete drevnom Šebezu da kreira mehanizam koji zadovoljava sledeće uslove:

- Kuglica se vraća u izvor posle nekoliko koraka.
- Kada se kuglica prvi put vrati u izvor, stanje svih razdelnika je 'X'.
- Kuglica se prvi put vraća u izvor nakon što je ušla u okidače tačno  $N$  puta. Serijski brojevi okidača, u redosledu kojim je kuglica ulazila u njih, moraju biti baš  $A_0, A_1, \dots, A_{N-1}$ .
- Označimo sa  $P$  ukupan broj promena stanja razdelnika koje je proizvelo kretanje kuglice do njenog prvog povratka u izvor. Vrednost  $P$  ne sme biti veća od 20 000 000.

Istovremeno, ne želite da koristite previše razdelnika u mehanizmu jer je poznato da ih je pravio Pavle.

## Detalji implementacije

Potrebno je implementirati sledeću funkciju.

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

- $M$ : broj okidača.
- $A$ : niz dužine  $N$  koji predstavlja serijske brojeve okidača kroz koje kuglica mora proći, u poretku kojim kuglica mora proći kroz njih.
- Ova funkcija se poziva tačno jednom.
- Obratite pažnju da je  $N$  dužina niza  $A$  i da se može dobiti na način kako je opisano u napomeni o implementaciji.

Vaš program treba da pozove sledeću funkciju kao odgovor.

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

- $C$ : niz dužine  $M + 1$ . Izlaz uređaja  $i$  ( $0 \leq i \leq M$ ) je povezan na uređaj  $C[i]$ .
- $X, Y$ : nizovi iste dužine. Dužina  $S$  oba ova niza je broj razdelnika. Za razdelnik  $-j$  ( $1 \leq j \leq S$ ), njegov izlaz 'X' je povezan na uređaj  $X[j - 1]$  a izlaz 'Y' je povezan na

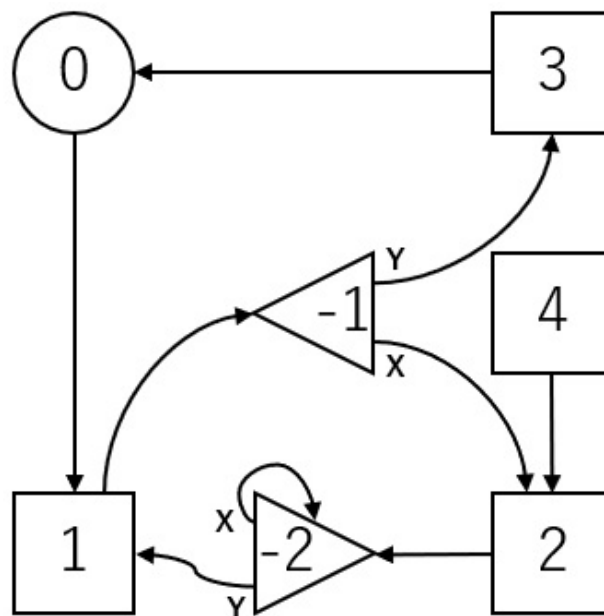
uređaj  $Y[j - 1]$ .

- Svaki element nizova  $C$ ,  $X$ , and  $Y$  mora biti ceo broj između  $-S$  i  $M$ , inkluzivno.
- $S$  ne sme biti veći od 400 000.
- Ova funkcija mora biti pozvana tačno jednom.
- Mehanizam koji je opisan nizovima  $C$ ,  $X$  i  $Y$  mora zadovoljavati uslove navedene u tekstu zadatka.

Ako neki od gore navedenih uslova nije zadovoljen, vaš program se ocenjuje kao **Wrong Answer**. Inače, vaš program se ocenjuje kao **Accepted** a vaši poeni se računaju na osnovu vrednosti  $S$  (pogledati deo Podzadaci).

## Primer

Neka je  $M = 4$ ,  $N = 4$ , i  $A = [1, 2, 1, 3]$ . Grejder poziva `create_circuit(4, [1, 2, 1, 3])`.



Slika iznad prikazuje mehanizam koji je opisan pozivom `answer([1, -1, -2, 0, 2], [2, -2], [3, 1])`. Brojevi na slici predstavljaju serijske brojeve uređaja.

Upotrebljena su dva razdelnika pa je  $S = 2$ .

Na početku, stanja razdelnika sa serijskim brojevima  $-1$  i  $-2$  su 'X'.

Kuglica putuje na sledeći način:

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

- Kada kuglica prvi put uđe u razdelnik  $-1$ , njegovo stanje je 'X', pa kuglica putuje ka okidaču 2. Zatim se stanje razdelnika  $-1$  menja na 'Y'.
- Kada kuglica drugi put uđe u razdelnik  $-1$ , njegovo stanje je 'Y', pa kuglica putuje ka okidaču 3. Zatim se stanje razdelnika  $-1$  menja na 'X'.

Kuglica se prvi put vraća u izvor nakon što je prošla kroz okidače 1, 2, 1, 3 (u tom redosledu). Oba razdelnika  $-1$  i  $-2$  se nalaze u stanju 'X'. Vrednost  $P$  je 4. Dakle, ovaj mehanizam zadovoljava sve uslove.

Fajl `sample-01-in.txt` u zipovanom dodatku odgovara ovom primeru. U zipovanom dodatku se nalaze i drugi primeri ulaza.

## Ograničenja

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

## Podzadaci

Poeni i ograničenja za svaki podzadatak su sledeća:

1. (2 poena) Za svako  $i$  ( $1 \leq i \leq M$ ), broj  $i$  se pojavljuje najviše jednom u nizu  $A_0, A_1, \dots, A_{N-1}$ .
2. (4 poena) Za svako  $i$  ( $1 \leq i \leq M$ ), broj  $i$  se pojavljuje najviše 2 puta u nizu  $A_0, A_1, \dots, A_{N-1}$ .
3. (10 poena) Za svako  $i$  ( $1 \leq i \leq M$ ), broj  $i$  se pojavljuje najviše 4 puta u nizu  $A_0, A_1, \dots, A_{N-1}$ .
4. (10 poena)  $N = 16$
5. (18 poena)  $M = 1$
6. (56 poena) Nema dodatnih ograničenja

Za svaki test primer, ako je vaš program ocenjen kao **Accepted**, broj poena se računa na osnovu vrednosti  $S$ :

- Ako je  $S \leq N + \log_2 N$ , dobijate sve poene predviđene za taj test primer.
- Za svaki test primer u Podzadacima 5 i 6, ako je  $N + \log_2 N < S \leq 2N$ , dobijate parcijalne poene. Broj poena za taj test primer je onda  $0.5 + 0.4 \times \left( \frac{2N - S}{N - \log_2 N} \right)^2$ , pomnožen brojem poena predviđenim za taj podzadatak.
- Inače, dobijate 0 poena.

Obratite pažnju da je broj poena na svakom od podzadataka jednak minimumu bodova na test primerima tog podzadatka.

## Priloženi grejder

Priloženi grejder učitava podatke sa standarnog ulaza u sledećem formatu:

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

Priloženi grejder štampa tri stvari.

Prvo, grejder štampa vaš odgovor u fajl `out.txt` u sledećem formatu.

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

Drugo, grejder simulira poteze kuglice tako što štampa serijske brojeve uređaja u koje je kuglica ulazila (redom) u fajl `log.txt`.

Treće, grejder štampa evaluaciju vašeg odgovora na standardni izlaz.

- Ako je vaš program ocenjen kao **Accepted**, priloženi grejder štampa brojeve  $S$  and  $P$  u formatu: `Accepted: S P`.
- Ako je program ocenjen sa **Wrong Answer**, štapa se: `Wrong Answer: MSG`. Značenje MSG je sledeće:
  - `answered not exactly once`: Procedura `answer` nije pozvana tačno jednom.
  - `wrong array length`: Dužina niza  $C$  nije  $M + 1$ , ili su dužine nizova  $X$  i  $Y$  različite.
  - `over 400000 switches`:  $S$  je veće od 400 000.
  - `wrong serial number`: Postoji element nizova  $C$ ,  $X$  ili  $Y$  koji je manji od  $-S$  ili veći od  $M$ .
  - `over 20000000 inversions`: Kuglica se ne vraća u izvor (origin) nakon 20 000 000 promena stanja razdelnika.
  - `state 'Y'`: Postoji razdelnik čije je stanje 'Y' kada se kuglica vrati u izvor.
  - `wrong motion`: Niz kojim su aktivirani okidači se razlikuje od niza  $A$ .

Obratite pažnju, priloženi grejder možda neće napraviti fajlove `out.txt` i/ili `log.txt` kada je vaš program ocenjen sa `Wrong Answer`.