



# Mechanical Doll

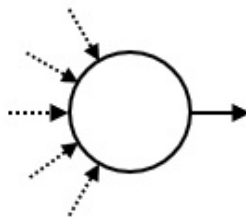
Lutke koje mogu automatizovano ponavljati specifični niz kretnji nazivamo mehaničkim lutkama. U Japanu su još od drevnih vremena konstruisane takve lutke.

Kretnje mehaničke lutke kontroliše **mehanizam** koji se sastoji od **uređaja (komponenti)**. Uređaji su međusobno povezani cijevima. Svaki uređaj ima jedan ili dva **izlaza**, i može imati nula ili više **ulaza**.

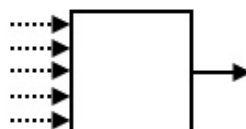
Svaka od cijevi povezuje izlaz jednog uređaja sa ulazom tog istog ili nekog drugog uređaja. Na svaki ulaz je povezana tačno jedna cijev i na svaki izlaz je povezana tačno jedna cijev.

Da bi opisali način na koji se lutka pokreće, razmotrimo **lopticu** koja se postavlja u neki od uređaja mehanizma. Loptica sada putuje kroz mehanizam. U svakom koraku putovanja, loptica napušta uređaj kroz neki od izlaza, kreće se kroz cijev koja je povezana na taj izlaz i ulazi u uređaj koji je na drugom kraju cijevi. Postoje tri tipa uređaja: **starter (origin)**, **okidač (trigger)** i **razdjelnik (switch)**. U mehanizmu postoji tačno jedan starter,  $M$  okidača i  $S$  razdjelnika ( $S$  može biti i nula). Vaš zadatak je da odlučite kolika je vrijednost  $S$ . Svaki od uređaja ima jedinstveni serijski broj.

Starter je uređaj u kojem se na početku postavlja loptica. Starter ima samo jedan izlaz i njegov serijski broj je 0.

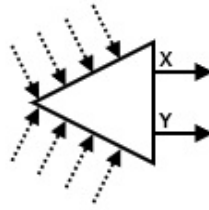


Okidač je uređaj koji omogućava da lutka izvrši jedan tip specifične kretnja kada god loptica uđe u okidač. Svaki okidač ima tačno jedan izlaz. Serijski brojevi okidača su od 1 do  $M$ .



Svaki razdjelnik ima dva izlaza, koje označavamo sa 'X' i 'Y'. **Stanje** može biti ili 'X' ili 'Y'. Kada loptica uđe u razdjelnik, izlazi iz njega na onaj izlaz koji je označen trenutnim stanjem razdjelnika. Po izlasku loptice, razdjelnik prelazi u suprotno stanje. Na

početku, svaki razdjelnik se nalazi u stanju 'X'. Serijski brojevi razdjelnika su negativni, od  $-1$  do  $-S$ .



Dat 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 kreirate mehanizam koji zadovoljava sljedeće uslove:

- Loptica se vraća u starter poslije nekoliko koraka.
- Kada se loptica prvi put vrati u starter, stanje svih razdjelnika je 'X'.
- Loptica se prvi put vraća u starter pošto je ušla u okidače tačno  $N$  puta. Uzastopni serijski brojevi okidača, u poretku kojim je loptica ulazila u njih, dati su sa  $A_0, A_1, \dots, A_{N-1}$ .
- Označimo sa  $P$  ukupan broj promjena stanja razdjelnika koje je proizvelo kretanje loptice do njenog prvog povratka u starter. Vrijednost  $P$  ne smije biti veća od 20 000 000.

Sa druge strane, izrada razdjelnika je složena, pa ne želite veliki broj razdjelnika u mehanizmu.

## Detalji implementacije

Potrebno je implementirati sljedeću funkciju

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

- $M$ : broj okidača.
- $A$ : niz dužine  $N$  koji predstavlja serijske brojeve okidača kroz koje loptica mora proći, u poretku kojim loptica u njih ulazi.
- 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 obavještenjima.

Vaš program treba da pozove sljedeću 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 razdjelnika. Za razdjelnik  $-j$  ( $1 \leq j \leq S$ ), njegov izlaz 'X' je povezan na uređaj  $X[j - 1]$  a izlaz 'Y' je povezan

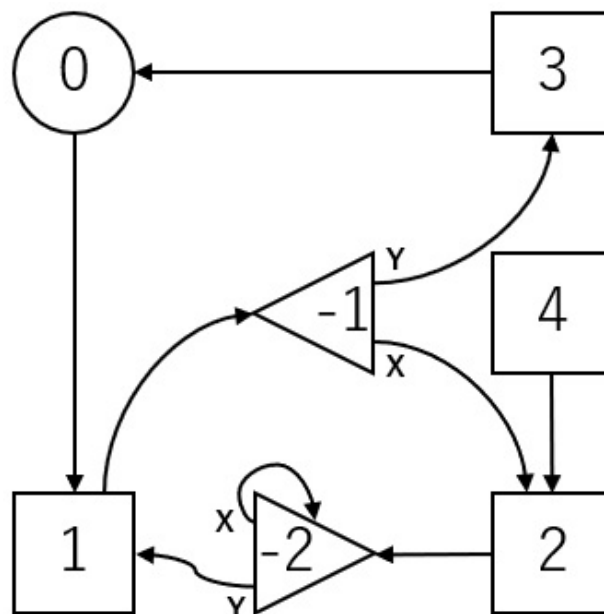
na uređaj  $Y[j - 1]$ .

- Svaki od elementa nizova  $C$ ,  $X$  i  $Y$  mora biti cio broj između  $-S$  i  $M$ , uključivo.
- $S$  ne smije 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 dobija ocjenu **Wrong Answer** U suprotnom, vaš program dobija ocjenu **Accepted** a vaši bodovi se računaju na osnovu vrijednosti  $S$  (pogledati sekciju **Podzadaci** za detalje).

## Primjer

Pretpostavimo da je  $M = 4$ ,  $N = 4$  i  $A = [1, 2, 1, 3]$ . Program za ocjenjivanje (grader) poziva `create_circuit(4, [1, 2, 1, 3])`.



Na slici je prikazan mehanizam koji se dobija pozivom `answer([1, -1, -2, 0, 2], [2, -2], [3, 1])`. Brojevi na slici predstavljaju serijske brojeve uređaja.

Upotrebljena su dva razdjelnika, pa je  $S = 2$ .

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

Putanja loptice je sljedeća:

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

- Kad loptica prvi put uđe u razdjelnik  $-1$ , njegovo stanje je 'X', pa loptica putuje ka okidaču 2. Stanje razdjelnika  $-1$  je promijenjeno na 'Y'.
- Kada loptica po drugi put uđe u razdjelnik  $-1$ , njegovo stanje je 'Y', pa loptice kreće ka okidaču 3. Stanje razdjelnika  $-1$  je promijenjeno na 'X'.

Loptica se prvi put vraća u starter i prošla je kroz okidače 1, 2, 1, 3. Oba razdjelnika  $-1$  i  $-2$  se nalaze u stanju 'X'. Vrijednost  $P$  je 4. Dakle, ovaj mehanizam zadovoljava uslove zadatka.

U zip-fajlu priloženom uz ovaj zadatak, fajl `sample-01-in.txt` opisuje ovaj primjer. Zip-fajlu sadrži i druge primjere 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

Bodovi i ograničenja za svaki test primjer:

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

Za svaki test primjer, ako je ocjena za vaš program **Accepted**, broj bodova se izračunava na osnovu vrijednosti  $S$ :

- Ako je  $S \leq N + \log_2 N$ , dobijate sve bodove predviđene za test primjer.
- Za svaki od test primjera u podzadacima 5 i 6, ako je  $N + \log_2 N < S \leq 2N$ , dobijate samo djelimične bodove. Broj bodova za svaki test se izračunava po formuli  $0.5 + 0.4 \times \left( \frac{2N - S}{N - \log_2 N} \right)^2$  i množi se brojem bodova predviđenim za podzadatak.
- U suprotnom, broj bodova je 0.

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

## Primjer programa za ocjenjivanje (sample grader)

Program za ocjenjivanje koji se nalazi u zip-fajlu priloženom uz zadatak učitava podatke u sljedećem formatu.

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

Program za ocjenjivanje koji se nalazi u zip-fajlu priloženom uz zadatak daje tri izlaza.

Prvo, program za ocjenjivanje štampa vaš odgovor u fajl `out.txt` u sljedećem formatu.

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

Drugo, program za ocjenjivanje simulira kretanje loptice. Štampa serijske brojeve uređaja kroz koje je loptica prošla, u poretku kojim ih je prolazila, u fajl `log.txt`.

Treće, program za ocjenjivanje štampa ocjenu vašeg odgovora na standardni izlaz.

- Ako je vaš program ocijenjen sa **Accepted**, program za ocjenjivanje štampa `iP$` su sljedećm formatu `Accepted: S P`.
- Ako je vaš program ocijenjen sa **Wrong Answer**, štampa se `Wrong Answer: MSG`. Značenja MSG može biti:
  - `answered not exactly once`: Funkcija `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`: U nekom od nizova `C`, `X` ili `Y` postoji element koji je manji od  $-S$  ili veći od  $M$ .
  - `over 20000000 inversions`: Loptica se nije vratila u starter za najviše 20 000 000 promjena stanja razdjelnika.
  - `state 'Y'`: Postoji razdjelnik čije je stanje, poslije prvog povratka loptice u starter, jednako 'Y'.
  - `wrong motion`: Okidači koji su uzrok kretnji se razlikuju od niza `A`.

Obratite pažnju da u slučaju kada je vaš program ocijenjen sa `Wrong Answer`, program za ocjenjivanje (grader) možda neće kreirati neki od fajlova `out.txt` i `log.txt`.