



Mehāniskā lelle

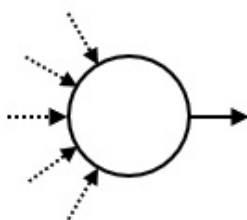
Mehāniskā lelle ir mehānisms, kas automātiski atkārtu kādu noteiktu kustību secību. Japānā mehāniskās lelles ir veidotas kopš seniem laikiem.

Mehāniskās lelles kustības kontrolē **shēma**, kas sastāv no **ierīcēm**. Ierīces ir savienotas ar caurulēm. Katrai ierīcei var būt neierobežoti liels skaits **ieeju** (var arī nebūt neviena), un viena vai divas **izejas**. Katra caurule savieno kādas ierīces izeju ar tās pašas vai citas ierīces ieeju. Tieši viena caurule ir pievienota katrai ieejai un tieši viena caurule ir pievienota katrai izejai.

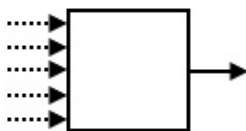
Lai paskaidrotu kā lelle izdara kustības, iedomājieties ka vienā no ierīcēm ir ievietota **bumbiņa**. Tad bumbiņa sāk ceļot pa shēmu. Katrā ceļojuma solī bumbiņa aizceļo no ierīces pa vienu no izejām, tālāk pa atbilstošo cauruli ieceļo ierīcē, kas atrodas šīs caurules otrā galā (iespējams, tajā pašā ierīcē).

Ir trīs ierīču veidi: **sākumierīce**, **trigeris** un **slēdzis**. Ir tikai viena sākumierīce, M triggeru, un S slēdžu (S var būt nulle). Jums ir jānosaka S vērtība. Katrai ierīcei ir unikāls seriāls numurs.

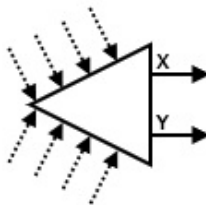
Sākumierīce ir ierīce kur sākotnēji atrodas bumbiņa. Tai ir viena izeja. Tās seriālais numurs ir 0.



Trigeris liek lellei izdarīt vienu noteikta tipa kustību kad bumbiņa tajā ieceļo. Katram triggerim ir viena izeja. Triggeru seriālie numuri ir no 1 līdz M .



Katram slēdzim ir divas izejas, kuras apzīmētas ar 'X' un 'Y'. Slēdža **stāvoklis** ir vai nu 'X', vai 'Y'. Kad bumbiņa ieceļo slēdzī, tā izceļo no tā izmantojot izeju, kas atbilst tekošajam slēdža stāvoklim. Pēc tam slēdzis maina savu stāvokli uz pretējo. Sākotnēji, katrs slēdzis ir stāvoklī 'X'. Slēdžu seriālie numuri ir no -1 līdz $-S$.



Jums ir dots trigeru skaits M . Jums arī ir dota virkne A garumā N , kur katrs tās elements ir kāda trigeru seriālais numurs. Katrs trigeris virknē A var parādīties vairākas (iespējams, nevienu) reizes. Jūsu uzdevums ir izveidot shēmu, kas atbilst šādiem noteikumiem:

- Bumbiņai jāatgriežas sākumierīcē pēc kāda soļu skaita.
- Kad bumbiņa pirmoreiz atgriezīsies sākumierīcē, katram slēdzim ir jābūt stāvoklī 'X'.
- Bumbiņai pirmo reizi jāatgriežas sākumierīcē tad, kad tā ir ieceļojusi trigeros tieši N reizes. Seriālo numuru secībai kādā bumbiņa ierodas trigeros jābūt A_0, A_1, \dots, A_{N-1} .
- Ar P apzīmēsim kopējo slēdžu stāvokļu izmaiņu skaitu, kas noticis pirms bumbiņa pirmo reizi atgriežas sākumierīcē. P vērtība nedrīkst pārsniegt 20 000 000.

Jūs vēlaties izmantot pēc iespējas mazāk slēdžu.

Implementācijas detaļas

Jums ir jāimplementē šāda procedūra:

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

- M : trigeru skaits.
- A : masīvs garumā N , kas apraksta trigeru seriālos numurus tādā secībā, kādā bumbiņai tajos jāierodas.
- Šī procedūra tiks izsaukta tieši vienreiz.
- Pievērsiet uzmanību, ka vērtība N ir masīva A garums, un var tikt iegūts kā norādīts implementācijas norādījumos.

Lai iesniegtu savu atbildi, jūsu programmai ir jāizsauc šāda procedūra.

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

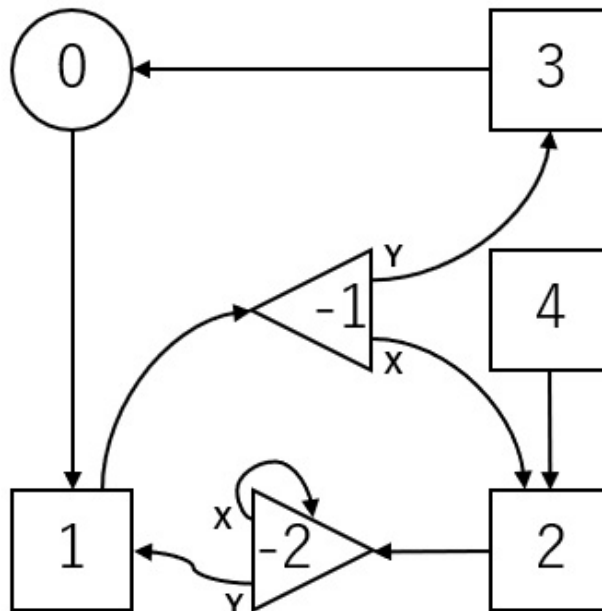
- C : masīvs garumā $M + 1$, kas raksturo sākumierīci un trigerus. Ierīces ar seriālo numuru i ($0 \leq i \leq M$) izeja ir savienota ar ierīces $C[i]$ ieeju.
- X, Y : masīvi ar vienādu garumu. Šo masīvu garums S ir slēdžu skaits. Slēdža ar seriālo numuru $-j$ ($1 \leq j \leq S$) izeja 'X' ir savienota ar ierīces $X[j - 1]$, bet izeja 'Y' - ar ierīces $Y[j - 1]$ ieeju.

- Visiem C , X , un Y elementiem ir jābūt veseliem skaitļiem no $-S$ līdz M , ieskaitot.
- S vērtība nedrīkst pārsniegt 400 000.
- Šī procedūra ir jāizsauc tieši vienreiz.
- Shēmai, ko apraksta masīvi C , X , un Y , jāievēro visi uzdevuma aprakstā minētie nosacījumi.

Ja kāds no augstākminētajiem nosacījumiem neizpildīsies, jūsu programma ir novērtēta ar **Wrong Answer**. Pretējā gadījumā, jūsu programma tiks novērtēta ar **Accepted** un jūsu punktu skaits tiks aprēķināts atkarībā no S vērtības (skatīt *Apakšuzdevumi*).

Piemērs

Pieņemsim, ka $M = 4$, $N = 4$, un $A = [1, 2, 1, 3]$. Vērtētājs izsauc `create_circuit(4, [1, 2, 1, 3])`.



Šajā diagrammā parādīta shēma, kas atbilst izsaukumam `answer([1, -1, -2, 0, 2], [2, -2], [3, 1])`. Skaitļi diagrammā ir ierīču seriālie numuri.

Ir izmantoti divi slēdži. Tātad, $S = 2$.

Sākotnēji, slēdži -1 un -2 atrodas stāvoklī 'X'.

Bumbiņa ceļo šādi:

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

- Kad bumbiņa pirmo reizi ieceļo slēdžī ar seriālo numuru -1 , tas atrodas stāvoklī 'X'. Tātad, bumbiņa izceļo uz trigeri ar seriālo numuru 2. Pēc tam slēdža -1 stāvoklis mainās uz 'Y'.
- Kad bumbiņa ieceļo slēdžī ar seriālo numuru -1 otro reizi, tas atrodas stāvoklī 'Y'. Tātad, bumbiņa izceļo uz trigeri ar seriālo numuru 3. Pēc tam slēdža -1 stāvoklis

mainās uz 'X'.

Kad bumbiņa pirmo reizi atgriežas sākumierīcē, tā triggerus ir apmeklējusi secībā 1, 2, 1, 3. Slēdži ar seriālajiem numuriem -1 un -2 atrodas stāvoklī 'X'. P vērtība ir 4. Tātad šī shēma atbilst nosacījumiem.

Files `sample-01-in.txt` pievienotajā pakotnes arhīvā atbilst tikko aprakstītajam piemēram. Arhīvā ir pieejami arī citu piemēru ievaddati.

Ierobežojumi

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

Apakšuzdevumi

Punktu skaits un ierobežojumi katram testam ir šādi:

1. (2 punkti) Katram i ($1 \leq i \leq M$), skaitlis i parādas virknē A_0, A_1, \dots, A_{N-1} ne vairāk kā vienreiz.
2. (4 punkti) Katram i ($1 \leq i \leq M$), skaitlis i parādas virknē A_0, A_1, \dots, A_{N-1} ne vairāk kā divas reizes.
3. (10 punkti) Katram i ($1 \leq i \leq M$), skaitlis i parādas virknē A_0, A_1, \dots, A_{N-1} ne vairāk kā četras reizes.
4. (10 punkti) $N = 16$
5. (18 punkti) $M = 1$
6. (56 punkti) Bez papildu ierobežojumiem.

Katram testam, ja jūsu programma ir novertēta ar **Accepted**, jūsu punktu skaits tiks aprēķināts atkarībā no S vērtības šādi:

- Ja $S \leq N + \log_2 N$, jūs par šo testu saņemsiet pilnu punktu skaitu.
- Katram 5. un 6. apakšuzdevuma testam, ja $N + \log_2 N < S \leq 2N$, jūs saņemsiet daļēju punktu skaitu. Punktu skaits par šo testu ir vienāds ar $0.5 + 0.4 \times \left(\frac{2N - S}{N - \log_2 N} \right)^2$, sareizinot to ar pilnu punktu skaitu par šo testu.
- Citādi, punktu skaits ir 0.

Ievērojiet, ka punktu skaits katrā apakšuzdevumā ir mazākais punktu skaits, kāds iegūts šī apakšuzdevuma testos.

Paraugvērtētājs

Paraugvērtētājs ielasa ievaddatus no standarta ievada plūsmas šādā formātā:

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

Paraugvērtētājs ģenerē trīs izvadus.

Pirmkārt, paraugvērtētājs izvada jūsu atbildi failā ar nosaukumu `out.txt` šādā formātā:

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

Otrkārt, paraugvērtētājs simulē bumbiņas kustību. Tas izvada ierīču seriālos numurus bumbiņas ieceļošanas secībā failā ar nosaukumu `log.txt`.

Treškārt, paraugvērtētājs izvada jūsu atbildes vērtējumu standarta izvada plūsmā.

- Ja jūsu programma ir novērtēta ar **Accepted**, tad paraugvērtētājs drukā S un P formātā: `Accepted: S P`.
- Ja jūsu programma ir novērtēta ar **Wrong Answer**, tad paraugvērtētājs drukā `Wrong Answer: MSG`. `MSG` nozīme ir šāda:
 - `answered not exactly once`: Procedūra `answer` tika izsaukta vairāk vai mazāk nekā vienu reizi.
 - `wrong array length`: Masīva `C` garums nav $M + 1$, vai masīvu `X` un `Y` garumi atšķiras.
 - `over 400000 switches`: S ir lielāks par 400 000.
 - `wrong serial number`: Masīvā `C`, `X`, vai `Y` eksistē elements kurš ir mazāks par $-S$ vai lielāks par M .
 - `over 20000000 inversions`: Bumbiņa neatgriežas sākumierīcē pēc 20 000 000 slēdžu stāvokļu maiņām.
 - `state 'Y'`: Kad bumbiņa pirmo reizi atgriežas sākumierīcē, eksistē slēdzis kurš atrodas stāvoklī 'Y'.
 - `wrong motion`: Trigeru secība, kādos tajos ieceļo bumbiņa, atšķiras no virknes A .

Pievērsiet uzmanību ka paraugvērtētājs var neizveidot `out.txt` un/vai `log.txt` failus, ja jūsu programma ir novērtēta ar `Wrong Answer`.