



Mechanická panenka

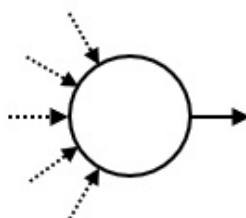
Mechanická panenka je starojaponský automat, který opakuje určitou posloupnost pohybů.

Pohyby panenky kontroluje **obvod**, který se skládá z **udělátek**. Udělátka jsou propojena trubkami. Každé udělátko může mít libovolný (i nulový) počet **vchodů** a jeden nebo dva **východy** dle typu udělátka. Každá trubka spojuje východ udělátka se vchodem stejného nebo jiného udělátka. Ke každému vchodu i východu je připojena právě jedna trubka.

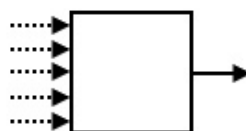
Pohyby panenky ovládá **kulička**, která se kutálí obvodem. Je-li v udělátku, vykutálí se jedním z výstupů (určeným dle typu udělátka) a připojenou trubkou se prokutálí do udělátka na jejím druhém konci. Kulička se nikdy nezastaví.

K dispozici máte tři typy udělátek: právě jedno **startovní**, předepsaný počet M **kontrolních** a libovolný vámi určený (případně i nulový) počet S **rozdvojek**. Každé udělátko má jednoznačné sériové číslo.

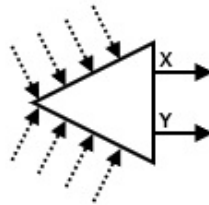
Startovní udělátko má právě jeden východ a je v něm na začátku běhu zařízení umístěná kulička. Sériové číslo startovního udělátka je 0.



Kontrolní udělátko pohne panenkou předepsaným způsobem kdykoliv se do něj vkutálí kulička. Kontrolní udělátko má právě jeden východ. Kontrolní udělátka mají sériová čísla 1 až M .



Rozdvojka má právě dva východy, označené 'X' a 'Y'. Rozdvojka je vždy v jednom ze dvou stavů, také označených 'X' a 'Y'. Kulička se z rozdvojky vykutálí východem označeným stejně jako aktuální stav. Poté se rozdvojka přepne do druhého ze stavů. Počáteční stav všech rozdvojek je 'X'. Sériová čísla rozdvojek jsou -1 až $-S$.



Máte předepsán počet kontrolních udělátek M a posloupnost A délky N , skládající se ze sériových čísel (ne nutně všech) kontrolních udělátek. Vytvořte obvod splňující následující podmínky:

- Po nějakém konečném počtu kroků se kulička vrátí do startovního udělátka.
- Když se do něj poprvé vrátí, stav všech rozdvojek je 'X'.
- Kulička se do startovního udělátka vrátí poprvé poté, co vstoupí do kontrolních udělátek právě N -krát, postupně do kontrolních udělátek sériových čísel A_0, A_1, \dots, A_{N-1} .
- Nechť P je celkový počet změn stavů rozdvojek předtím, než se kulička poprvé vrátí do startovního udělátka. Hodnota P nesmí být větší než 20 000 000.

Přitom se snažte použít co nejméně rozdvojek.

Implementační detaily

Implementujte následující proceduru.

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

- M : počet kontrolních udělátek.
- A : pole délky N udávající posloupnost sériových čísel kontrolních udělátek v pořadí, v němž se do nich má vkutálet kulička.
- Tato procedura bude zavolána právě jednou.
- Hodnota N je rovna délce pole A a můžete ji určit postupem popsáním na papíru Poznámky k implementaci.

Váš program dá odpověď zavoláním následující procedury.

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

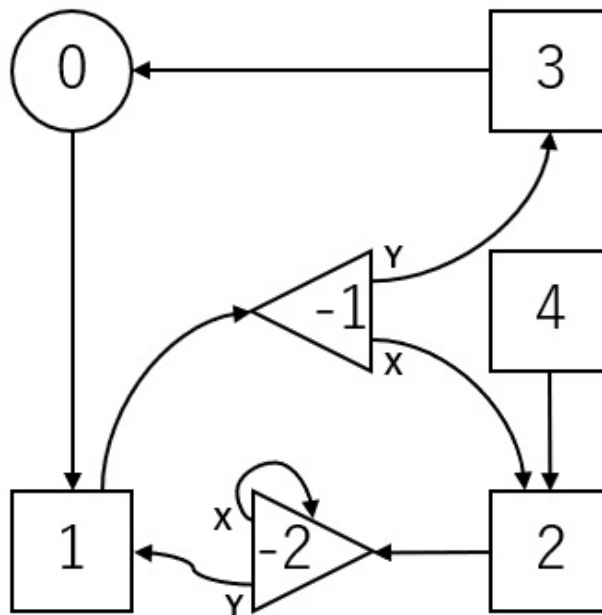
- C : pole délky $M + 1$. Východ udělátka se sériovým číslem i ($0 \leq i \leq M$) je spojen trubkou se vchodem udělátka $C[i]$.
- X, Y : pole stejné délky S udávající počet rozdvojek. Rozdvojka se sériovým číslem $-j$ ($1 \leq j \leq S$) má východ 'X' spojen trubkou se vchodem udělátka $X[j - 1]$ a východ 'Y' se vchodem udělátka $Y[j - 1]$.
- Čísla v polích C, X a Y musí být mezi $-S$ a M včetně.
- S musí být nejvýše 400 000.

- Tuto proceduru musíte zavolat právě jednou.
- Obvod popsany poli C , X a Y musí splňovat podmínky zadání.

Jestliže některá z těchto podmínek není splněna, váš program bude vyhodnocen jako **Wrong Answer**. Jinak váš program bude vyhodnocen jako **Accepted** a počet bodů bude přidělen dle hodnoty S způsobem popsany v sekci Podúlohy.

Příklad

Nechť $M = 4$, $N = 4$ a $A = [1, 2, 1, 3]$. Vyhodnocovač zavolá `create_circuit(4, [1, 2, 1, 3])`.



Váš program může popsat například výše zobrazený obvod (čísla v obrázku jsou sériová čísla udělatek), zavoláním `answer([1, -1, -2, 0, 2], [2, -2], [3, 1])`.

V obvodu jsou dvě rozdvojky, a proto $S = 2$.

Na začátku běhu jsou rozdvojky -1 a -2 ve stavu 'X'. Kulička se obvodem kutálí následovně:

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

- Když se kulička poprvé vkutálí do rozdvojky se sériovým číslem -1 , stav rozdvojky je 'X'. Proto se kulička prokutálí do udělatek 2. Poté se stav rozdvojky -1 změní na 'Y'.
- Když se kulička vkutálí do rozdvojky se sériovým číslem -1 podruhé, stav rozdvojky je 'Y'. Kulička se proto prokutálí do udělatek 3. Poté se stav rozdvojky -1 změní na 'X'.

Kulička se vrátí do startovního udělatek poprvé poté, co postupně navštívila kontrolní udělatek 1, 2, 1, 3. Stav rozdvojek -1 a -2 je v tomto okamžiku 'X'. Hodnota P je 4.

Obvod tedy splňuje předepsané podmínky.

Soubor `sample-01-in.txt` v zazipovaném archívu odpovídá tomuto příkladu. V archívu najdete i další vzorové vstupy.

Omezení

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

Podúlohy

Bodování a podmínky jednotlivých podúloh jsou následující:

1. (2 body) Pro každé i ($1 \leq i \leq M$) se číslo i v posloupnosti A_0, A_1, \dots, A_{N-1} vyskytne nejvýše jednou.
2. (4 body) Pro každé i ($1 \leq i \leq M$) se číslo i vyskytne v posloupnosti A_0, A_1, \dots, A_{N-1} nejvýše dvakrát.
3. (10 bodů) Pro každé i ($1 \leq i \leq M$) se číslo i vyskytne v posloupnosti A_0, A_1, \dots, A_{N-1} nejvýše čtyřikrát.
4. (10 bodů) $N = 16$
5. (18 bodů) $M = 1$
6. (56 bodů) Bez dalších omezení.

Je-li výstup vašeho programu vyhodnocen jako **Accepted**, počet získaných bodů je určen následovně dle hodnoty S :

- Jestliže $S \leq N + \log_2 N$, získáte plný počet bodů dle specifikace podúlohy.
- Jestliže $N + \log_2 N < S \leq 2N$, v podúlohách 5 a 6 získáte $0.5 + 0.4 \times \left(\frac{2N - S}{N - \log_2 N} \right)^2$ krát počet bodů dle specifikace podúlohy.
- Jinak získáte 0 bodů.

Počet bodů za každou podúlohu je minimum z počtů bodů získaných za testovací vstupy v této podúloze.

Ukázkový testovač

Ukázkový testovač čte vstup ze standardního vstupu v následujícím formátu.

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

Ukázkový testovač produkuje tři výstupy.

Nejprve vypíše odpověď vašeho programu do souboru `out.txt` v následujícím formátu.

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

Poté simuluje chování obvodu a postupně vypisuje sériová čísla udělátek, kterými se kutálí kulička, do souboru `log.txt`.

Nakonec vypíše na standardní výstup vyhodnocení vašeho programu.

- Je-li program vyhodnocen **Accepted**, vzorový testovač vypíše hodnoty S a P ve formátu `Accepted: S P`.
- Je-li program vyhodnocen **Wrong Answer**, vypíše `Wrong Answer: MSG`. Význam zprávy `MSG` je následující:
 - `answered not exactly once`: Proceduru `answer` jste zavolali víc než jednou.
 - `wrong array length`: Délka pole C není $M + 1$ nebo pole X a Y mají různé délky.
 - `over 400000 switches`: S je větší než 400 000.
 - `wrong serial number`: Jedno z polí C , X a Y obsahuje číslo menší než $-S$ nebo větší než M .
 - `over 20000000 inversions`: Proběhlo víc než 20 000 000 změn stavů rozdvojek a kulička se zatím nevrátila do startovního udělátka.
 - `state 'Y'`: Když se kulička vrátila do startovního udělátka, některá z rozdvojek je ve stavu 'Y'.
 - `wrong motion`: Kulička neprošla kontrolními udělátky v pořadí předepsaném posloupností A .

Když je váš program vyhodnocen `Wrong Answer`, vyhodnocovač nemusí nutně vytvořit soubory `out.txt` či `log.txt`.