Mikroshēma

Mikroshēma sastāv no $N + M$ **vārtiem**, kas sanumurēti no 0 līdz $N + M - 1$. Vārti no 0 līdz $N - 1$ ir **sliekšņa vārti**, bet vārti no N līdz $N + M - 1$ ir **avota vārti**.

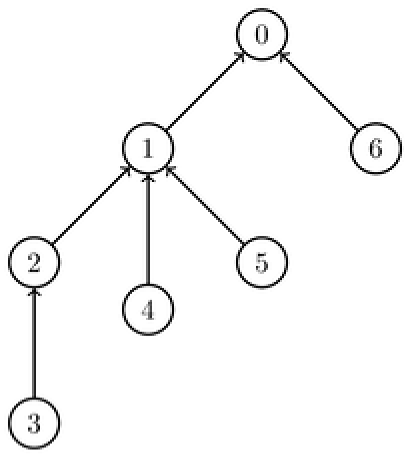
Katri vārti, izņemot vārtus 0, ir **ievads** tieši vieniem sliekšņa vārtiem. Precīzāk, katram i, kur $1 \le i \le N + M - 1$, vārti i ir ievads vārtiem $P[i]$, kur $0 \le P[i] \le N - 1$. Svārīgi, ka ir spēkā arī $P[i] < i$. Turklāt mēs pieņemsim, ka $P[0] = -1$. Katriem sliekšņa vārtiem ir viens vai vairāki ievadi. Avota vārtiem nav neviena ievada.

Katriem vārtiem ir **stāvoklis** kas ir vai nu 0, vai 1. Avota vārtu sākotnējais stāvoklis ir noteikts ar masīva A, kas sastāv no M skaitļiem, elementu vērtībām. Tas ir, katram j, kur $0 \le j \le M - 1$, avota vārtu $N + j$ sākotnējs stāvoklis ir $A[j]$.

Katru sliekšņa vārtu stāvoklis ir atkarīgs no to ievadu stāvokļiem un ir noteikts sekojoši: Pirmkārt, katram sliekšņa vārtiem ir piešķirts sliekšņa **parametrs**. Parametram, kas ir piešķirts sliekšņa vārtiem ar c ievadiem, ir jābūt skaitlim starp 1 un c (ieskaitot). Tad, sliekšņa vārtu ar parametru p stāvoklis ir 1, ja vismaz p tā ievadiem ir stāvoklis 1, un 0 pretējā gadījumā.

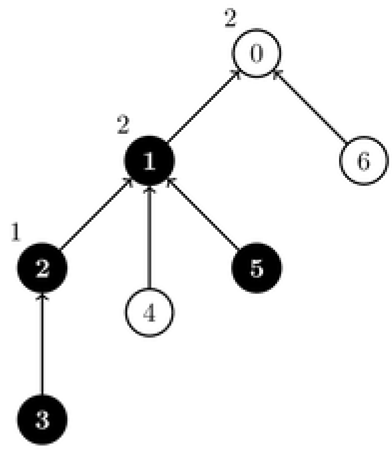
Pieņemsim, ka ir $N = 3$ sliekšņu vārti un $M = 4$ avota vārti. Vārtu 0 ievadi ir vārti 1 un 6, vārtu 1 ievadi ir vārti 2, 4 un 5, un vārtu 2 vienīgais ievads ir vārti 3.

Šīs piemērs ir parādīts attēlā:



Pieņemsim ka avota vārtu 3 un 5 stāvoklis ir 1, bet avota vārtiem 4 un 6 stāvoklis ir 0. Pieņemsim, ka mēs piešķirsim parametrus 1, 2 un 2 attiecīgi sliekšņa vārtiem 2, 1 un 0. Šajā gadījumā vārtiem 2

stāvoklis ir 1, vārtiem 1 stāvoklis ir 1 un vārtiem 0 stāvoklis ir 0. Šī parametru vērtību piešķire un vārtu stāvokļi ir illustrēti sekojošā attēlā. Vārti kuru stāvoklis ir 1, ir apzīmēti ar melnu krāsu.



Avota vārtu stāvokļi tiks atjaunināti Q reizes. Katru atjauninājumu apraksta divi skaitļi L un R ($N \le L \le R \le N + M - 1$) un nomaina stāvokļus uz pretējiem visiem avotu vārtiem, kuru numuri ir no L līdz R (ieskaitot). Tas ir, katram i, kur $L \le i \le R$, avotu vārtu i stāvoklis mainās uz 1, ja to stāvoklis bija 0; vai uz 0, ja to stāvoklis bija 1. Jaunais stāvoklis katriem nomainītiem vārtiem paliek nemainīgs līdz tas varbūt tiks nomainīts kādā vēlākā atjauninājumā.

Jūsu uzdevums pēc katra atjauninājuma ir saskaitīt, cik ir dažādu veidu kā piešķirt parametru vērtības visiem sliekšņa vārtiem, lai nodrošinātu, ka vārtu 0 stāvoklis ir 1. Divi veidi kā piešķirt parametru vērtību visiem sliekšņa vārtiem ir dažādi, ja eksistē vismaz vieni sliekšņa vārti, kam abos veidos ir atšķirīgi stāvokļi. Tā kā veidu skaits var būt liels, jums ir jāizrēķina tas pēc moduļa $1\;000\;002\;022$.

Ņemiet vērā, kā augstākminētajā piemērā, ir 6 dažādi veidi kā piešķirt parametru vērtību visiem sliekšņa vārtiem, jo vārtiem 0, 1 un 2 ir attiecīgi 2, 3 un 1 ievads. Divos no šiem 6 veidiem, vārtiem 0 būs stāvoklis 1.

Realizācijas detaļas

Ir jārealizē šādas procedūras:

```
void init(int N, int M, int[] P, int[] A)
```

- N: sliekšņa vārtu skaits.
- M: avota vārtu skaits.
- P: masīvs garumā $N + M$, kas apraksta sliekšņu vārtu ievadus.
- A: masīvs garumā M, kas apraksta avotu vārtu sakotnējus stāvokļus.
- Šī procedūra tiek izsaukta tieši vienreiz, pirms jebkuriem `count_ways` izsaukumiem.

```
int count_ways(int L, int R)
```

- L, R: ierobežojumi avotu vārtu diapazonam, kuru stāvokļi tiek mainīti uz pretējo.
- Šai procedūrai sākumā ir jāizdara aprakstītais atjauninājums un tad jāatgriež dažādo veidu skaits (pēc moduļa $1\ 000\ 002\ 022$) kā piešķirt parametru vērtību visiem sliekšņa vārtiem, lai nodrošinātu, ka vārtiem 0 stāvoklis ir 1.
- Šī procedūra tiks izsaukta tieši Q reizes.

Piemērs

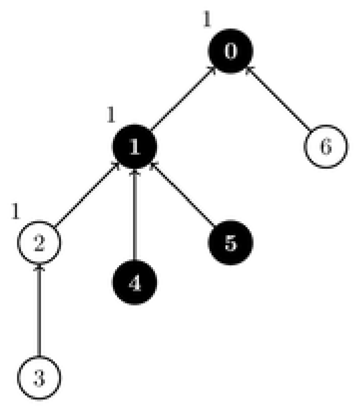
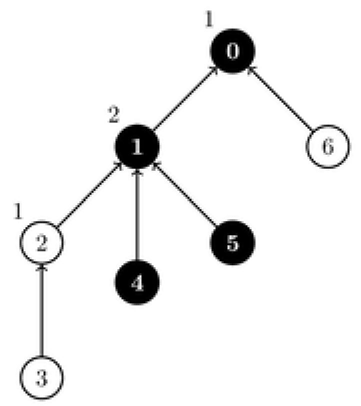
Aplūkosim šādu izsaukumu secību:

```
init(3, 4, [-1, 0, 1, 2, 1, 1, 0], [1, 0, 1, 0])
```

Šīs piemērs ir ilustrēts uzdevuma aprakstā augstāk.

```
count_ways(3, 4)
```

Šis nomaina stāvokļus uz pretējiem vārtiem 3 un 4, tas ir vārtu 3 stāvoklis kļūst 0, un vārtu 4 stāvoklis kļūst 1. Divi veidi kā var piešķirt parametru vērtību visiem sliekšņa vārtiem, lai nodrošinātu, ka vārtiem 0 stāvoklis ir 1, ir ilustrēti attēlos zemāk.

Pirmais veids	**Otrais veids**
	

Visās pārējās parametru vērtību piešķiršanās, vārtiem 0 stāvoklis ir 0. Tātad, procedūrai ir jāatgriež 2.

```
count_ways(4, 5)
```

Šīs nomaina stāvokļus uz pretējiem vārtiem 4 un 5. Rezultātā, visiem avotu vārtiem stāvoklis ir 0, un, jebkurai parametru piešķiršanai, vārtu 0 stāvoklis ir 0. Tātad, procedūrai ir jāatgriež 0.

```
count_ways(3, 6)
```

Šīs nomaina visu vārtu stāvokļus uz 1. Rezultātā, jebkurai parametru piešķiršanai, vārtu 0 stāvoklis ir 1. Tātad, procedūrai ir jāatgriež 6.

Ierobežojumi

- $1 \leq N, M \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $P[0] = -1$
- $0 \leq P[i] < i$ and $P[i] \leq N - 1$ (katram i, kur $1 \leq i \leq N + M - 1$)
- Katriem sliekšņa vārtiem ir vismaz viens ievads (katram i, kur $0 \leq i \leq N - 1$, eksistē indekss x ka $i < x \leq N + M - 1$ un $P[x] = i$).
- $0 \leq A[j] \leq 1$ (katram j, kur $0 \leq j \leq M - 1$)
- $N \leq L \leq R \leq N + M - 1$

Apakšuzdevumi

1. (2 punkti) $N = 1$, $M \leq 1000$, $Q \leq 5$
2. (7 punkti) $N, M \leq 1000$, $Q \leq 5$, katriem sliekšņa vārtiem it tieši divi ievadi.
3. (9 punkti) $N, M \leq 1000$, $Q \leq 5$
4. (4 punkti) $M = N + 1$, $M = 2^z$ (kādam pozitīvam skaitlim z), $P[i] = \lfloor \frac{i-1}{2} \rfloor$ (katram i, kur $1 \leq i \leq N + M - 1$), $L = R$
5. (12 punkti) $M = N + 1$, $M = 2^z$ (kādam pozitīvam skaitlim z), $P[i] = \lfloor \frac{i-1}{2} \rfloor$ (katram i, kur $1 \leq i \leq N + M - 1$)
6. (27 punkti) Katriem sliekšņa vārtiem it tieši divi ievadi.
7. (28 punkti) $N, M \leq 5000$
8. (11 punkti) Bez papildu ierobežojumiem.

Paraugvērtētājs

Paraugvērtētājs lasa ievaddatus šādā formātā:

- 1. rinda: $N \; M \; Q$
- 2. rinda: $P[0] \; P[1] \; \ldots \; P[N + M - 1]$
- 3. rinda: $A[0] \; A[1] \; \ldots \; A[M - 1]$
- $(4 + k)$-tā rinda ($0 \leq k \leq Q - 1$): $L \; R$ (atjauninājumam k)

Paraugvērtētājs izvada jūsu atbildes šādā formātā:

- $(1+k)$-tā rinda $(0 \le k \le Q-1)$: procedūras `count_ways` atgrieztā vērtība k-ajam atjauninājumam