Digitalno vezje

Imamo digitalno vezje, ki je sestavljeno iz $N + M$ **vrat**, ki so oštevilčene od 0 do $N + M - 1$. Vrata med 0 in $N - 1$ so **univerzalna vrata**, medtem ko so vrata od N do $N + M - 1$ **vhodna vrata**.

Vsaka vrata, razen vrat 0, so vhod v natanko ena univerzalna vrata. Natančneje, za vsak i, tako da je $1 \le i \le N + M - 1$, so vrata i vhod v vrata $P[i]$, kjer 0 \le P[i] \le N-1$. Pomembno je, da velja tudi P [i] < i.

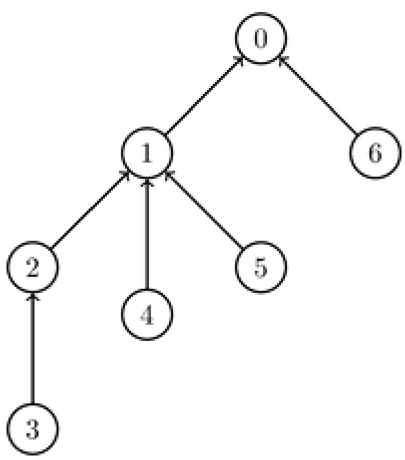
Poleg tega predpostavimo, da velja $P[0] = -1$. Vsaka univerzalna vrata imajo enega ali več vhodov. Vhodna vrata nimajo nobenih vhodov.

Vsaka vrata so lahko v **stanju** 0 ali 1. Začetna stanja vhodnih vrat so podana s poljem A, ki vsebuje M celih števil. To pomeni, da je za vsak j, kjer velja $0 \le j \le M - 1$, začetno stanje vodnih vrat $N + j$ is $A[j]$.

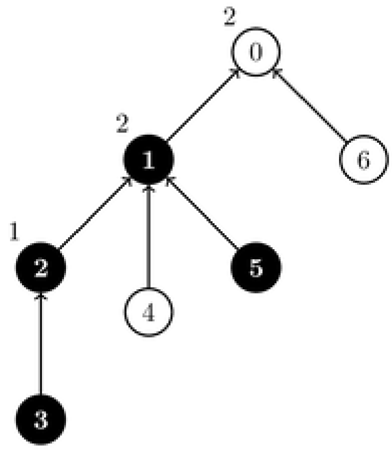
Stanje vsakih univerzalnih vrat je odvisno od stanj njegovih vhodov in je določeno na naslednji način. Prvič, vsakim univerzalnim vratom je dodeljen univerzalni **parameter**. Parameter, dodeljen univerzalnim vratom s c vhodi, mora biti celo število med 1 in c (vključno). Nadalje je stanje univerzalnih vrat s parametrom p enako 1, če ima vsaj p njegovih vhodov stanje 1, sicer je stanje 0.

Na primer, predpostavimo, da imamo $N = 3$ univerzalnih vrat in $M = 4$ vhodnih vrat. Vhodi v vrata 0 so vrata 1 in 6, vhodi v vrata 1 so vrata 2, 4, in 5, edini vhod v vrata 2 pa so vrata 3.

Opisan primer je prikazan na naslednji sliki.



Recimo, da imata vhodna vrata 3 in 5 stanje 1, medtem ko imata vhodni vrati 4 in 6 stanje 0. Predpostavimo, da dodelimo parametre 1, 2 in 2 univerzalnim vratom 2, 1 in 0 (v tem vrstnem redu). V tem primeru imajo vrata 2 stanje 1, vrata 1 stanje 1 in vrata 0 stanje 0. Ta dodelitev vrednosti parametrov in stanj je prikazana na naslednji sliki. Vrata, ki imajo stanje je 1, so označena s črno barvo.



Stanja vhodnih vrat bodo podvržena Q posodobitvam. Vsaka posodobitev je opisana z dvema celima številoma L in R ($N \leq L \leq R \leq N + M - 1$) in preklopi stanja vseh vhodnih vrat, oštevilčenih med vključno L in R. To pomeni, da za vsak i, kjer velja $L \leq i \leq R$, vhodna vrata i spremenijo svoje stanje v 1, če je njihovo stanje 0, ali v 0, če je njihovo stanje 1. Novo stanje vsakih preklopnih vrat ostane nespremenjeno, dokler ni je morda preklopljena z eno od poznejših posodobitev.

Tvoja naloga je prešteti, po vsaki posodobitvi, koliko razilčnih dodelitev parametrov pragovnim vratom povzroči, da imajo vrata 0 stanje 1. Dve dodelitvi se štejeta za različni, če obstaja vsaj ena univerzalna vrata, ki imajo drugačno vrednost parametra v obeh prireditvah. Ker je število načinov lahko zelo veliko, ga morate izračunati po modulu $1\;000\;002\;022$.

Upoštevajte, da je v zgornjem primeru 6 različnih dodelitev parametrov univerzalnim vratom, saj imajo vrata 0, 1 in 2 število vhodov 2, 3 and 1. V 2 od teh 6 dodelitev imajo vrata 0 stanje 1.

Podrobnosti implementacije

Vaša naloga je napisati proceduro in funkcijo.

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

- N: število univerzalnih vrat.
- M: število vhodnih vrat.
- P: polje dolžine $N + M$, ki hrani vhode v univerzalna vrata.
- A: polje dolžine M, ki hrani podatke o začetnih stanjih vhodnih vrat.

- Ta procedura se pokliče natančno enkrat, pred kakršnimi koli klici `count_ways`.

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

- L, R: meje zaporedja vhodnih vrat, katerih stanja se preklapljajo.
- Ta postopek mora najprej izvesti predvideno posodobitev, nato pa vrne število načinov modulo $1\;000\;002\;022$ dodeljevanja parametrov univerzalnim vratom, ki ima za posledico, da imajo vrata 0 stanje 1.
- Ta funkcijak se pokliče točno Q-krat.

Primer

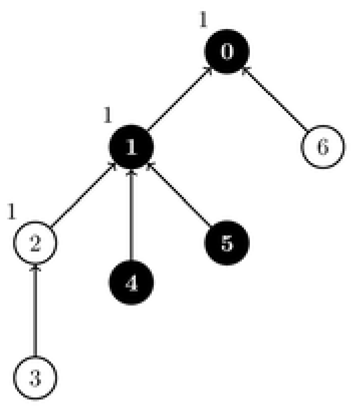
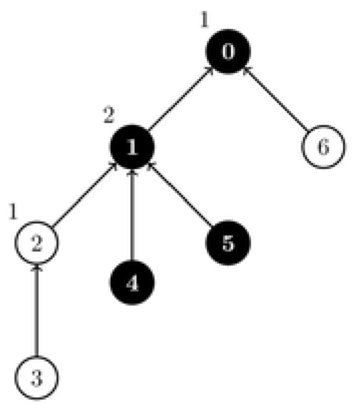
Recimo, da imamo naslednje zaporedje klicev:

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

Ta primer je prikazan zgoraj v opisu naloge.

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

To preklopi stanja vrat 3 in 4, tj. stanje vrat 3 postane 0, stanje vrat 4 pa 1. Na spodnjih slikah sta prikazana dva načina dodeljevanja parametrov, zaradi katerih imajo vrata 0 stanje 1.

Način 1	**Način 2**
	

Pri vseh drugih dodelitvah parametrov imajo vrata 0 stanje 0. Zato mora procedura vrniti vrednost 2.

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

Ta klic povzroči preklop stanj vrat 4 in 5. Posledično imajo vsa vhodna vrata stanje 0 in za vsako dodelitev parametrov imajo vrata 0 stanje 0. Zato mora procedura vrniti vrednost 0.

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

Ta klic spremeni stanja vseh vhodnih vrat na 1. Posledično imajo vrata 0 stanje 1 za katero koli dodelitev parametrov. Zato mora procedura vrniti vrednost 6.

Omejitve

- $1 \leq N, M \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $P[0] = -1$
- $0 \leq P[i] < i$ in $P[i] \leq N - 1$ (za vsak i, tako da je $1 \leq i \leq N + M - 1$)
- vsaka univerzalna vrata imajo najmanj en vhod (za vsak i, tako da velja $0 \leq i \leq N - 1$ obstaja indeks x, tako da $i < x \leq N + M - 1$ in $P[x] = i$).
- $0 \leq A[j] \leq 1$ (za vsak j, tako da je $0 \leq j \leq M - 1$)
- $N \leq L \leq R \leq N + M - 1$

Podnaloge

1. (2 točki) $N = 1$, $M \leq 1000$, $Q \leq 5$
2. (7 točk) $N, M \leq 1000$, $Q \leq 5$, vsaka univerzalna vrata imajo natančno dva vhoda.
3. (9 točk) $N, M \leq 1000$, $Q \leq 5$
4. (4 točke) $M = N + 1$, $M = 2^z$ (za neko pozitivno celo število z), $P[i] = \lfloor \frac{i-1}{2} \rfloor$ (za vsak i, tako da je $1 \leq i \leq N + M - 1$), $L = R$
5. (12 točk) $M = N + 1$, $M = 2^z$ (za neko pozitivno celo število z), $P[i] = \lfloor \frac{i-1}{2} \rfloor$ (za vsak i, tako da je $1 \leq i \leq N + M - 1$)
6. (27 točk) Vsaka univerzalna vrata imajo natančno dva vhoda.
7. (28 točk) $N, M \leq 5000$
8. (11 točk) Brez dodatnih omejitev.

Vzorčni ocenjevalnik

Vzorčni ocenjevalnik bere vhod naslednje oblike:

- vrstica 1: $N \; M \; Q$
- vrstica 2: $P[0] \; P[1] \; \ldots \; P[N + M - 1]$
- vrstica 3: $A[0] \; A[1] \; \ldots \; A[M - 1]$
- vrstica $4 + k$ ($0 \leq k \leq Q - 1$): $L \; R$ za posodobitev k

Vzorčni ocenjevalnik izpiše odgovore v naslednji obliki:

- vrstica $1 + k$ ($0 \leq k \leq Q - 1$): vrnjena vrednost funkcije `count_ways` za posodobitev k