Digitális áramkör

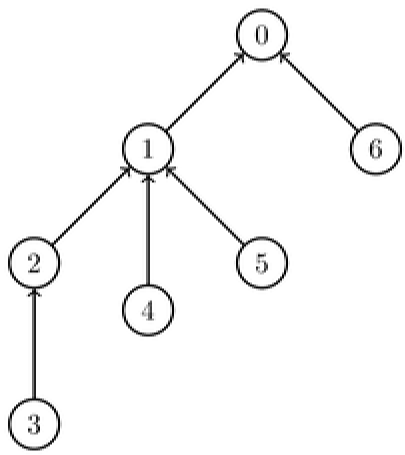
Adott egy áramkör, amelyben $N + M$ **kapu** van, 0-tól $N + M - 1$-ig sorszámozva. A 0-tól $N - 1$-ig sorszámozott kapuk **küszöbkapuk**, míg az N-től $N + M - 1$-ig sorszámozottak **forráskapuk**.

A 0. kaput kivéve, minden kapu pontosan egy küszöbkapunak **bemenete**. Konkrétabban, minden i-re, ahol $1 \le i \le N + M - 1$, az i. kapu a $P[i]$. kapunak egy bemenete, ahol $0 \le P[i] \le N - 1$. Fontos továbbá, hogy $P[i] < i$. Ezen kívül kikötjük, hogy $P[0] = -1$. Minden küszöbkapunak egy vagy több bemenete van. A forráskapuknak nincs bemenete.

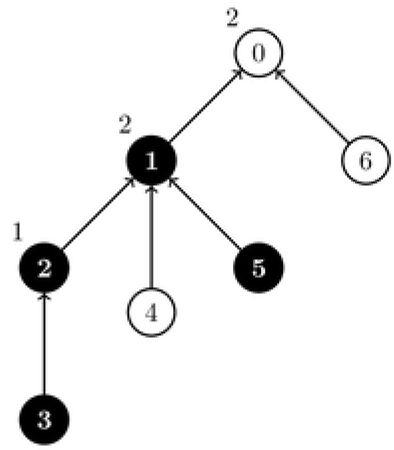
Minden kapunak van egy **állapota**, ami 0 vagy 1 lehet. A forráskapuk kezdeti állapotai egy M elemű A tömbben vannak megadva. Vagyis minden j-re, ahol $0 \le j \le M - 1$, az $N + j$. forráskapu állapota $A[j]$.

Minden küszöbkapu állapotát a bemeneteinek állapota határozza meg a következőképpen. Először minden küszöbkapunak megadunk egy küszöb**paramétert**. Egy c darab bemenettel rendelkező küszöbkapu paramétere csak egy 1 és c közötti egész szám lehet (a határokat beleértve). Majd, egy p paraméterű küszöbkapu állapota 1 lesz, ha a bemenetei közül legalább p-nek az állapota 1, egyébként pedig 0.

Például, tegyük fel, hogy $N = 3$ küszöbkapu van, és $M = 4$ forráskapu. A 0. kapu bemenetei az 1. és a 6. kapu, az 1. kapu bemenetei a 2., a 4. és az 5. kapu, míg a 2. kapu egyetlen bemenete a 3. kapu. Ezt a példát a következő ábra szemlélteti.



Tegyük fel, hogy a 3. és 5. forráskapuk állapota 1, míg a 4. és 6. forráskapuk állapota 0. Tegyük fel, hogy az 1, 2 és 2 értékeket adjuk meg paraméterként rendre a 2., 1. és 0. küszöbkapunak. Ebben az esetben, a 2. kapu állapota 1, az 1. kapu állapota 1 és a 0. kapu állapota 0. Ezt a paraméterkiosztást és a hozzá tartozó állapotokat az alábbi ábra szemlélteti. Feketével vannak jelölve az 1 állapotú kapuk.



A forráskapuk állapotain Q változtatás történik. Egy változtatást két egész szám ír le, L és R ($N \leq L \leq R \leq N+M-1$), és hatására átkapcsolódik az összes L és R közötti (a határokat beleértve) sorszámú forráskapu állapota. Vagyis, minden i-re, ahol $L \leq i \leq R$, az i. forráskapu állapota 1-re változik, ha 0 volt az állapota, illetve 0-ra, ha 1 volt az állapota. Minden átkapcsolt kapu állapota változatlan marad addig, amíg esetleg egy későbbi változtatás átkapcsolja.

A feladatod minden változtatás után kiszámítani, hogy hány különböző paraméterkiosztás esetén lesz a 0. kapu állapota 1. Két kiosztás különböző, ha legalább egy küszöbkapu paraméterértéke más a két kiosztásban. Mivel az eredmény nagy szám is lehet, az $1\ 000\ 002\ 022$-vel vett maradékát kell megadnod.

Megjegyezzük, hogy a fenti példában 6 különböző kiosztása lehet a küszöbkapuk paramétereinek, hiszen a 0., 1. és 2. kapunak rendre 2, 3 és 1 bemenete van. A 6 kiosztás közül 2 esetén lesz a 0. kapu állapota 1.

Megvalósítás

Két függvényt kell elkészítened.

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

- N: a küszöbkapuk száma.
- M: a forráskapuk száma.
- P: egy $N+M$ elemű tömb, ami a küszöbkapuk bemeneteit írja le.
- A: egy M elemű tömb, ami a forráskapuk kezdeti állapotait adja meg.
- Ezt a függvényt pontosan egyszer hívják, a `count_ways` függvény bármely hívása előtt.

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

- L, R: az átkapcsolt forráskapuk intervallumának határai.
- Ez a függvény a változtatás elvégzése után az olyan paraméterkiosztások számát adja vissza, modulo $1\ 000\ 002\ 022$, amelyek esetén a 0. kapu állapota 1!
- Ezt a függvényt pontosan Q-szor hívják.

Példa

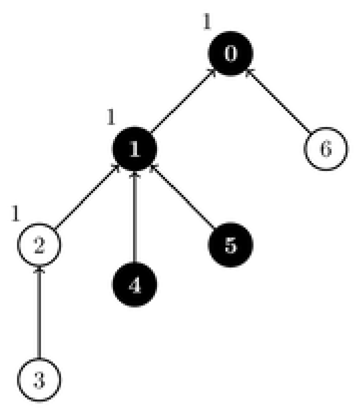
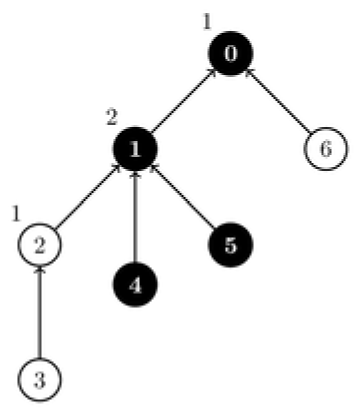
Tekintsük a függvényhívások alábbi sorát:

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

Ez a példa a feladatleírásban van szemléltetve.

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

Ez a 3. és a 4. kapu állapotát kapcsolja át, tehát a 3. kapu állapota 0 lesz, és a 4. kapu állapota 1 lesz. Két olyan paraméterkiosztás van, amely esetén a 0. kapu állapota 1, az alábbi ábra szerint.

1. eset	**2. eset**
	

Minden más paraméterkiosztás esetén a 0. kapu állapota 0. Tehát a függvénynek a 2 értéket kell visszaadnia.

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

Ez a 4. és az 5. kapu állapotát kapcsolja át. Ennek eredményeképpen minden forráskapu állapota 0 lesz, és bármely paraméterkiosztás esetén a 0. kapu állapota 0. Tehát a függvénynek a 0 értéket kell visszaadnia.

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

Ez minden forráskapu állapotát 1-re változtatja. Ennek eredményeképpen bármely paraméterkiosztás esetén a 0. kapu állapota 1 lesz. Tehát a függvénynek a 6 értéket kell visszaadnia.

Korlátok

- $1 \leq N, M \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $P[0] = -1$
- $0 \leq P[i] < i$ és $P[i] \leq N - 1$ (minden i-re, ahol $1 \leq i \leq N + M - 1$)
- Minden küszöbkapunak legalább egy bemenete van (vagyis minden i-re, ahol $0 \leq i \leq N - 1$ létezik egy olyan x sorszám, melyre $i < x \leq N + M - 1$ és $P[x] = i$).
- $0 \leq A[j] \leq 1$ (minden j-re, ahol $0 \leq j \leq M - 1$)
- $N \leq L \leq R \leq N + M - 1$

Részfeladatok

1. (2 pont) $N = 1$, $M \leq 1000$, $Q \leq 5$
2. (7 pont) $N, M \leq 1000$, $Q \leq 5$, minden küszöbkapunak pontosan két bemenete van.
3. (9 pont) $N, M \leq 1000$, $Q \leq 5$
4. (4 pont) $M = N + 1$, $M = 2^z$ (ahol z egy pozitív egész szám), $P[i] = \lfloor\frac{i-1}{2}\rfloor$ (minden i-re, ahol $1 \leq i \leq N + M - 1$), $L = R$
5. (12 pont) $M = N + 1$, $M = 2^z$ (ahol z egy pozitív egész szám), $P[i] = \lfloor\frac{i-1}{2}\rfloor$ (minden i-re ahol $1 \leq i \leq N + M - 1$)
6. (27 pont) Minden küszöbkapunak pontosan két bemenete van.
7. (28 pont) $N, M \leq 5000$
8. (11 pont) Nincsenek további korlátok.

Mintaértékelő

A mintaértékelő a standard bemenetről a következő formátumban olvas be:

- 1. sor: $N \ M \ Q$
- 2. sor: $P[0] \ P[1] \ \ldots \ P[N + M - 1]$
- 3. sor: $A[0] \ A[1] \ \ldots \ A[M - 1]$
- $4 + k$. sor ($0 \leq k \leq Q - 1$): $L \ R$ a k. változtatás paraméterei

A mintaértékelő az alábbi formátumban írja ki a válaszaidat:

- $1 + k$. sor ($0 \leq k \leq Q - 1$): a `count_ways` visszatérési értéke a k. változtatásra