Circuito Digital

Hay un circuito, que consiste de $N+M$ **puertas** numeradas desde 0 hasta $N+M-1$. Las puertas desde 0 hasta $N-1$ son puertas de umbral, mientras que las puertas desde N hasta $N+M-1$ son puertas de origen.

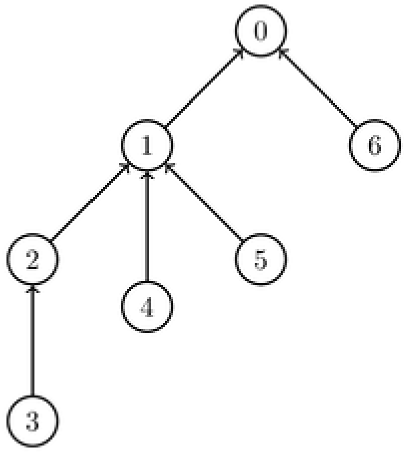
Cada puerta, excepto la puerta 0, es una **entrada** para exactamente una puerta de umbral. Específicamente, por cada i tal que $1 \leq i \leq N+M-1$, la puerta i es una entrada para la puerta $P[i]$, donde $0 \leq P[i] \leq N-1$. Es importante destacar que $P[i] < i$. Además, se debe suponer que $P[0] = -1$. Cada puerta de umbral tiene una o más entradas. Las puertas de origen no tienen entradas.

Cada puerta tiene un **estado** que puede ser 0 o 1. Los estados iniciales de las puertas de origen están dados en un arreglo A de M enteros. Esto es, para cada j tal que $0 \leq j \leq M-1$, el estado inicial de la puerta de origen $N+j$ es $A[j]$.

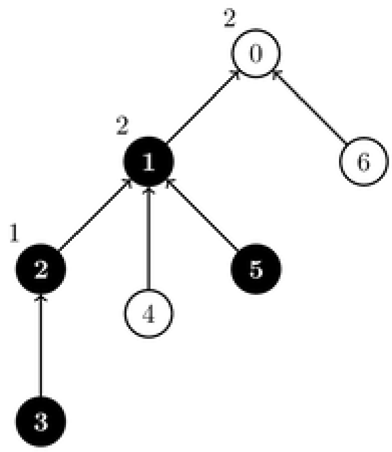
El estado de cada puerta de umbral depende del estado de sus entradas y se determina como sigue. Primero, cada puerta de umbral tiene asignado un **parámetro** de intercambio. El parámetro asignado a una puerta de umbral con c entradas debe ser un entero entre 1 y c (inclusive). Entonces, el estado de una puerta de umbral con parametro p es 1, si al menos una de sus entradas tiene estado 1, y 0 en caso contrario.

Por ejemplo, suponga que tenemos $N=3$ puertas de umbral y $M=4$ puertas de origen. Las entradas de la puerta 0 son las puertas 1 y 6, las entradas de la puerta 1 son las puertas 2, 4, y 5, y la unica entrada para la puerta 2 es la puerta 3.

Este ejemplo está ilustrado en la siguiente figura:



Supongamos que las puertas de origen 3 y 5 tienen estado 1, mientras las puertas de origen 4 y 6 tienen estado 0. Supongamos que asignamos los parámetros 1, 2 y 2 a las puertas de umbral 2, 1 y 0 respectivamente. En este caso, la puerta 2 tiene el estado 1, la puerta 1 tiene el estado 1 y la puerta 0 tiene el estado 0. Esta asignación de valores de parámetros y estados se ilustra en la siguiente imagen. Las puertas cuyo estado es 1 están marcadas en negro.



Los estados de la puerta de origen sufriran Q actualizaciones. Cada actualizacion es descrita por dos enteros L y R ($N \le L \le R \le N + M - 1$) y cambia los estados de todas las puertas de origen numeradas entre L y R, inclusive. Esto es, por cada i tal que $L \le i \le R$, la puerta de origen i cambia su estado a 1, si su estado es 0, o cambia a 0, si su estado es 1. El nuevo estado de cada puerta de origen permanece sin cambios hasta que posiblemente sea cambiado por una de las actualizaciones posteriores.

Su objetivo es contar, después de cada actualización, cuántas asignaciones diferentes de parámetros a las puertas de umbral dan como resultado que la puerta 0 tenga el estado 1. Dos asignaciones se consideran diferentes si existe al menos una puerta de umbral que tiene un valor diferente de su parámetro en ambas asignaciones. Como el número de maneras puede ser grande, debe calcularlo módulo $1\ 000\ 002\ 022$.

Tenga en cuenta que en el ejemplo anterior, hay 6 diferentes asignaciones de parámetros a las puertas de umbral, ya que las puertas 0, 1 y 2 tienen entradas 2, 3 y 1 respectivamente. En 2 de estas asignaciones de 6, la puerta 0 tiene el estado 1.

Detalles de Implementación

Tu tarea es implementar dos procedimientos.

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

- N: el número de puertas de intercambio.
- M: el número de puertas de origen.
- P: arreglo de longitud $N + M$ describiendo las entradas de las puertas de umbral.
- A: arreglo de longitud M describiendo el estado inicial de la puertas de origen.
- Este procedimiento se llama exactamente una vez, antes de cualquier llamada a `count_ways`.

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

- L, R: los límites del rango de puertas de origen, cuyos estados se cambian.
- Este procedimiento debe primero realizar la actualización especificada, y luego retornar el numero de maneras, modulo $1\,000\,002\,022$, de asignar parametros de las puertas de umbral, cuyo resultado en la puerta 0 sea el estado 1.
- Este procedimiento es llamado exactamente Q veces.

Ejemplo

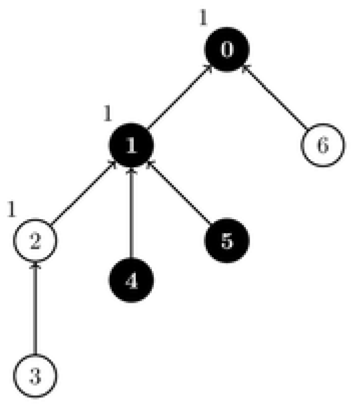
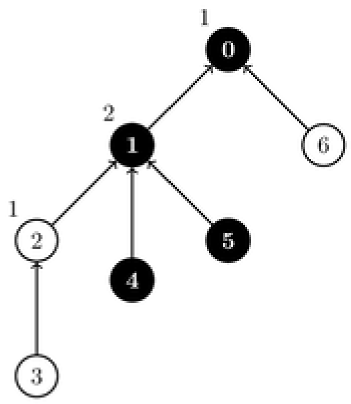
Considere la siguiente secuencia de llamadas:

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

Este ejemplo está ilustrado en la tarea descrita abajo.

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

Esto alterna los estados de las puertas 3 y 4, es decir, el estado de la puerta 3 se convierte en 0 y el estado de la puerta 4 se convierte en 1. En las siguientes imágenes se ilustran dos formas de asignar los parámetros que dan como resultado que la puerta 0 tenga el estado 1.

Forma 1	**Forma** 2
	

En todas las demás asignaciones de parámetros, la puerta 0 tiene el estado 0. Por lo tanto, el procedimiento debería devolver 2.

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

Esto cambia los estados de las puertas 4 y 5. Como resultado, todas las puertas de origen tienen el estado 0 y, para cualquier asignación de parámetros, la puerta 0 tiene el estado 0. Entonces el procedimiento debe retornar 0.

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

Esto cambia los estados de todas las puertas de origen a 1. Como resultado, para cualquier asignación de parámetros, la puerta 0 tiene el estado 1. Por lo tanto, el procedimiento debería devolver 6.

Restriccones

- $1 \leq N, M \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $P[0] = -1$
- $0 \leq P[i] < i$ y $P[i] \leq N - 1$ (para cada i tal que $1 \leq i \leq N + M - 1$)
- Cada puerta de umbral tiene al menos una entrada (para cada i tal que $0 \leq i \leq N - 1$ existe un índice x tal que $i < x \leq N + M - 1$ y $P[x] = i$)
- $0 \leq A[j] \leq 1$ (por cada j tal que $0 \leq j \leq M - 1$)
- $N \leq L \leq R \leq N + M - 1$

Subtareas

1. (2 puntos) $N = 1$, $M \leq 1000$, $Q \leq 5$
2. (7 puntos) $N, M \leq 1000$, $Q \leq 5$, cada puerta de umbral tiene exactamente dos entradas.
3. (9 puntos) $N, M \leq 1000$, $Q \leq 5$
4. (4 puntos) $M = N + 1$, $M = 2^z$ (para algún entero positivo z), $P[i] = \lfloor \frac{i-1}{2} \rfloor$ (para cada i tal que $1 \leq i \leq N + M - 1$), $L = R$
5. (12 puntos) $M = N + 1$, $M = 2^z$ (para algún entero positivo z), $P[i] = \lfloor \frac{i-1}{2} \rfloor$ (para cada i tal que $1 \leq i \leq N + M - 1$)
6. (27 puntos) Cada puerta de umbral tiene exactamente dos entradas.
7. (28 points) $N, M \leq 5000$
8. (11 points) Sin restricciones adicionales.

Evaluador de ejemplo

El evaluador de ejemplo lee las entredas de la siguiente forma:

- línea 1: $N \ M \ Q$
- línea 2: $P[0] \ P[1] \ \ldots \ P[N + M - 1]$
- línea 3: $A[0] \ A[1] \ \ldots \ A[M - 1]$
- línea $4 + k$ ($0 \leq k \leq Q - 1$): $L \ R$ para la actualización k.

El Evaluador de ejemplo imprime sus respuestas en el siguiente formato:

- línea $1 + k$ ($0 \leq k \leq Q - 1$): retorna el valor de `count_ways` para la actualización k.