Circuito Digital

Hay un circuito que consta de $N+M$ **compuertas** numeradas de 0 a $N+M-1$. Las compuertas 0 a $N-1$ son **compuertas de umbral**, mientras que las compuertas de N a $N+M-1$ son **compuertas de origen**.

Cada compuerta, excepto por la compuerta 0, es una **entrada** para exactamente una compuerta de umbral. Especificamente para cada i tal que $1 \leq i \leq N+M-1$, la compuerta i es una entrada de la compuerta $P[i]$, donde $0 \leq P[i] \leq N-1$. Es importante destacar también que se cumplirá que $P[i] < i$. Por otro lado, tomaremos $P[0] = -1$. Cada compuerta de umbral tiene una o más entradas. Las compuertas de origen no tienen ninguna entrada.

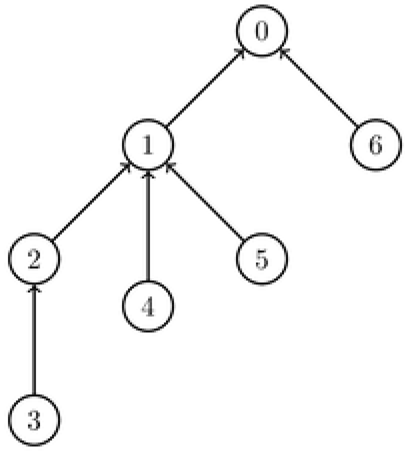
Cada compuerta tiene un **estado** que puede ser 0 o 1. Los estados iniciales de las compuertas de origen estarán dados por un arreglo A de M enteros. Es decir que para cada j tal que $0 \leq j \leq M-1$, el estado inicial de la compuerta de origen $N+j$ es $A[j]$.

El estado para cada compuerta de umbral depende de los estados de sus entradas y se determina de la siguiente forma.

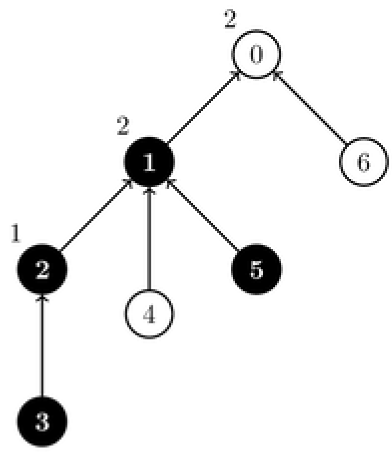
Primero, a cada compuerta de umbral le es asignado un **parámetro** de umbral. El parámetro asignado a una compuerta de umbral con c entradas debe ser un entero entre 1 y c (inclusive). Entonces el estado de la compuerta de umbral con el parámetro p es 1, si al menos p de sus entradas tienen estado 1, y será 0 en caso contrario.

Por ejemplo, supongamos que hay $N=3$ compuertas de umbral y $M=4$ compuertas de origen. Las entradas de la compuerta 0 son las compuertas 1 y 6, las entradas de la compuerta 1 son las compuertas 2, 4, y 5, y la única entrada de la compuerta 2 es la compuerta 3.

Este ejemplo está ilustrado en la siguiente figura.



Supongamos que las compuertas de origen 3 y 5 tienen estado 1, mientras que las compuertas de origen 4 y 6 tienen estado 0. Supongamos también que asignamos parámetros 1, 2 y 2 a las compuertas de umbral 2, 1 y 0 respectivamente. En este caso, la compuerta 2 tiene estado 1, la compuerta 1 tiene estado 1 y la compuerta 0 tiene estado 0. Esta combinación de valores de parámetros y estados se puede observar en la siguiente imagen. Las compuertas con estado 1 se muestran coloreadas de negro.



Los estados de las compuertas de origen se someterán a Q actualizaciones. Cada actualización se describe por dos enteros L y R ($N \leq L \leq R \leq N + M - 1$) e invierte los estados de todas las compuertas de origen numeradas de L a R inclusive. Esto significa que para cada i tal que $L \leq i \leq R$, la compuerta 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 compuerta invertida continuara sin cambios hasta que posiblemente sea invertida de nuevo por una de las actualizaciones futuras.

Tu objetivo es contar después de cada actualización, cuantas combinaciones distintas de parámetros asignados a las compuertas de umbral resultan en que la compuerta 0 tenga estado 1. Dos combinaciones se consideran distintas si exista al menos una compuerta umbral que tenga un diferente valor en sus parámetros entre las dos combinaciones. Como el número de maneras puede ser muy grande, tu debes calcularlo modulo $1\ 000\ 002\ 022$.

Observa qué en el ejemplo de arriba, existen 6 diferentes combinaciones de parámetros de compuertas de umbral, donde las compuertas 0, 1 y 2 tienen 2, 3 y 1 entradas respectivamente. En 2 de esas 6 combinaciones, la compuerta 0 tiene estado 1.

Detalles de Implementación

Tu tarea es implementar dos procedimientos.

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

- N: La cantidad de compuertas de umbral.
- M: La cantidad de compuertas de origen.
- P: Un arreglo de tamaño $N + M$ describiendo las entradas de las compuertas de umbral.
- A: Un arreglo de tamaño M describiendo los estados iniciales de las compuertas de origen.
- Este procedimiento es llamado exactamente una vez, antes de cualquier llamada a `count_ways`.

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

- L, R: los límites de los rangos de las compuertas de origen, cuyos estados se invertirán.
- Este procedimiento debe realizar primero la actualización especificada, y posteriormente regresar el número de maneras modulo $1\,000\,002\,022$, de asignar parámetros a las compuertas de umbral, que resulten en que la compuerta con estado 0 sea 1.
- Este procedimiento será llamado exactamente Q veces.

Ejemplo

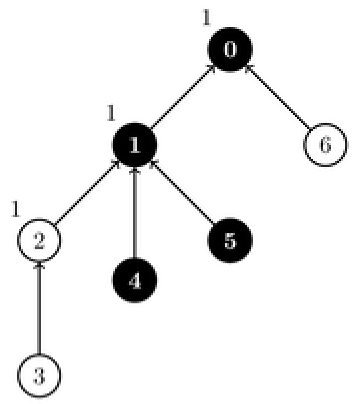
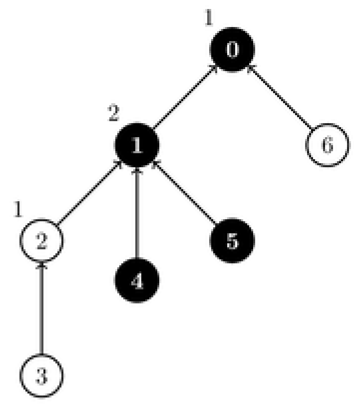
Considera la siguiente secuencia de llamadas:

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

Este ejemplo se puede observar arriba en la descripción de la tarea.

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

Esto invierte los estados de las compuertas 3 y 4, por lo tanto, el estado de la compuerta 3 se convierte en 0, y el estado de la compuerta 4 se convierte en 1. Existen dos formas de asignar los parámetros que dan como resultado que la compuerta 0 tenga estado 1, como se ilustra abajo en las imágenes.

Forma 1	**Forma** 2
	

En cualquier otra combinación de parámetros, la compuerta 0 tiene estado 0. Por lo tanto el procedimiento deberá regresar 2.

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

Esto invierte el estado de las compuertas de origen 4 y 5. Como resultado, todas las compuertas de origen tienen estado 0, y para cualquier combinación de parámetros, la compuerta 0 tendrá estado 0. Por lo tanto, el procedimiento deberá regresar 0.

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

Esto invierte el estado de todas las compuertas de entrada a 1. Como resultado para cualquier combinación de parámetros, la compuerta 0 tendrá estado 1. Por lo tanto, el procedimiento deberá regresar 6.

Restricciones

- $1 \leq N, M \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $P[0] = -1$
- $0 \leq P[i] < i$ y $P[i] \leq N - 1$ (para cada i tal que $1 \leq i \leq N + M - 1$)
- Cada compuerta de umbral tendrá al menos una entrada (para cada i tal que $0 \leq i \leq N - 1$ existirá un índice x tal que $i < x \leq N + M - 1$ y $P[x] = i$)
- $0 \leq A[j] \leq 1$ (para 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 compuerta 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 compuerta umbral tendrá exactamente dos entradas.
7. (28 puntos) $N, M \leq 5000$
8. (11 puntos) Sin restricciones adicionales.

Evaluador de Ejemplo

El evaluador de ejemplo lee la entrada en el siguiente formado:

- 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 k-ésima actualización

El evaluador de ejemplo imprimirá tus respuestas en el siguiente formato:

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