# Circuito Digital

Hay un circuito que consiste en $N+M$ **compuertas** enumeradas del $0$ al $N+M-1$. Las compuertas entre el $0$ y el $N-1$ son **compuertas de umbral**, mientras que las compuertas entre $N$ y $N+M-1$ son **compuertas de origen**.

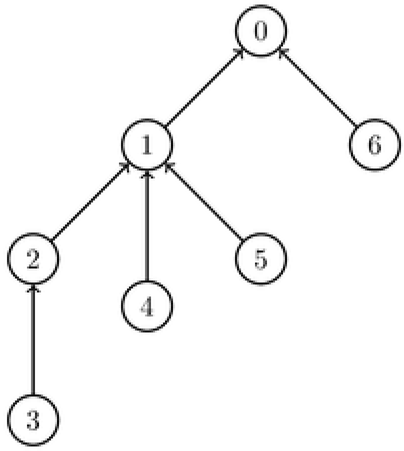
Cada compuerta, excepto por la compuerta $0$, es una **entrada** para exactamente una compuerta de umbral. Específicamente, para cada $i$ tal que $1 \le i \le N+M-1$, la compuerta $i$ es una entrada para la compuerta $P[i]$, donde $0 \le P[i] \le N-1$. Es importante recalcar que se tiene $P[i] < i$. Además, asumiremos que $P[0] = -1$. Cada compuerta de umbral tiene una o más entradas. Las compuertas de origen no tienen ninguna entrada.

Cada compuerta tiene un **estado**, el cual es $0$ o $1$. Los estados iniciales de las compuertas de origen serán dados por un vector $A$ de $M$ enteros. Esto es, para cada $j$ tal que $0 \le j \le M-1$, el estado inicial de la compuerta de origen $N+j$ es $A[j]$.

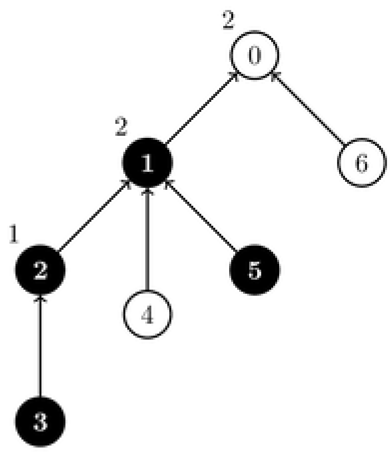
El estado de cada compuerta de umbral depende de los estados de sus entradas y es determinado de la siguiente manera: Primero, se asigna un **parámetro** de umbral a cada compuerta de umbral. El parámetro asignado a una compuerta de umbral que tiene $c$ entradas debe ser un entero entre $1$ y $c$ (inclusivo). Luego, el estado de una compuerta de umbral de parámetro $p$ es $1$ si al menos $p$ de sus entradas tienen estado $1$ o $0$ 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 que la única entrada de la compuerta $2$ es la compuerta $3$.

Este ejemplo está ilustrado en la siguiente imagen.




Supongamos que las compuertas de origen $3$ y $5$ tienen estado $1$, mientras que las compuertas de origen $4$ y $6$ tienen estado $0$. Asumamos que asignamos parámetros de $1$, $2$ y $2$ a las compuertas $2$, $1$ y $0$, respectivamente. En este caso, la compuerta $2$ tendría estado $1$, la compuerta $1$ tendría estado $1$ y la compuerta $0$ tendría estado $0$. Esta asignación de parámetros y estados está ilustrada en la siguiente imagen. Las compuertas cuyo estado sea $1$ están coloreadas de negro.




Los estados de las compuertas de origen recibirán $Q$ actualizaciones. Cada actualización está descrita por dos enteros $L$ y $R$ ($N \leq L \leq R \leq N + M - 1$) e invierte el estado de todas las compuertas de origen entre $L$ y $R$ (inclusivo). Esto es, 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 a $0$ si su estado es $1$. El nuevo estado de cada compuerta afectada se mantiene sin cambios hasta que sea invertido por alguna actualización posterior.

Su objetivo es contar, luego de cada actualización, cuántas asignaciones de parámetros diferentes resultan en que el estado de la compuerta $0$ sea $1$. Dos asignaciones son consideradas diferentes si existe al menos una compuerta con umbral que tenga diferente valor de parámetro en ambas asignaciones. Ya que la cantidad de asignaciones válidas puede ser muy grande, debe calcularla módulo $1\ 000\ 002\ 022$.

Note que en el anterior ejemplo hay $6$ asignaciones de parámetros diferentes, pues las compuertas $0$, $1$ y $2$ tienen $2$, $3$ y $1$ entradas, respectivamente. En $2$ de estas $6$ asignaciones, la compuerta $0$ tiene estado $1$.

## Detalles de Implementación

Su tarea es implementar dos funciones.

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

- $N$: Cantidad de compuertas de umbral.
- $M$: Cantidad de compuertas de origen.
- $P$: Un vector de longitud $N + M$ que describe la compuerta con umbral de la cual cada compuerta es entrada.
- $A$: Un vector de longitud $M$ que describe los estados iniciales de las compuertas de origen.
- Este función es llamada exactamente una vez, antes de cualquier llamada a `count_ways`.

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

- $L$, $R$: Los límites del rango de compuertas de origen cuyos estados serán invertidos.
- Esta función debe primero realizar la modificación especificada, y luego retorna el número de formas módulo $1\,000\,002\,022$, de asignar parámetros a las compuertas con umbral tales que resulten en que el estado de la compuerta $0$ sea $1$.
- Esta función es llamada exactamente $Q$ veces.

## Ejemplo

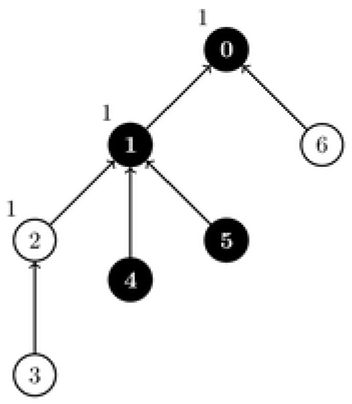
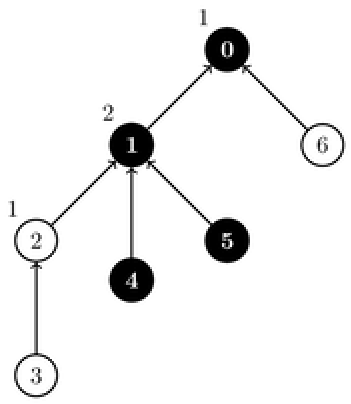
Considere la siguiente secuencia de llamadas.

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

Este ejemplo está ilustrado en la descripción del problema.

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

Esto invierte los estados de las compuertas $3$ y $4$; en otras palabras, el estado de la compuerta $3$ se vuelve $0$ y el estado de la compuerta $4$ se vuelve $1$. Dos maneras de asignar los parámetros que resulten en que el estado de la compuerta $0$ sea $1$ están ilustradas en las siguientes imágenes.

| **Manera** $1$ | **Manera** $2$ |
|---|---|
|   |   |

En todas las otras asignaciones de parámetros, la compuerta $0$ tiene estado $0$. Por lo tanto, la función debe devolver $2$.

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

Esto invierte los estados de las compuertas $4$ y $5$. Como resultado, todas las compuertas de origen tienen estado $0$, así que cualquier asignación de parámetros conllevará a que la compuerta $0$ tenga estado $0$. Por lo tanto, la función debe devolver $0$.

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

Esto cambia los estados de todas las compuertas de origen a $1$. Como resultado, cualquier asignación de parámetros conllevará a que la compuerta $0$ tenga estado $1$. Por lo tanto, la función debe devolver $6$.

## Restricciones

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

## Subtareas

1. (2 puntos) $N = 1$, $M \leq 1000$, $Q \leq 5$
2. (7 puntos) $N, M \leq 1000$, $Q \leq 5$, cada compuerta con 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 con umbral tiene exactamente dos entradas.
7. (28 puntos) $N, M \leq 5000$
8. (11 puntos) Sin restricciones adicionales.

## Grader de ejemplo

El grader de ejemplo lee la entrada con el siguiente formato:

- 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 grader de ejemplo imprime sus respuestas con el siguiente formato:

- línea $1 + k$ ($0 \leq k \leq Q - 1$): El valor que retorna `count_ways` luego de la $k$-ésima actualización.