Digital Circuit

Логическая схема состоит из $N + M$ **элементов**, пронумерованных от 0 до $N + M - 1$. Элементы с номерами 0 to $N - 1$ являются **пороговыми**, а элементы с номерами от N до $N + M - 1$ являются **стартовыми**.

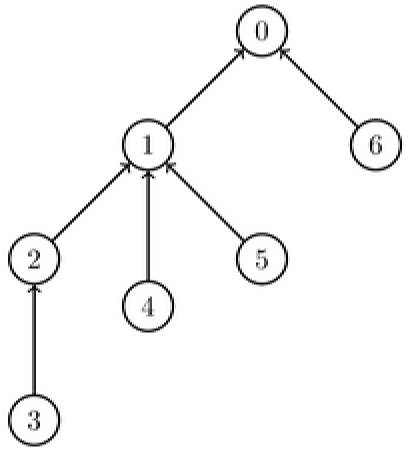
Каждый элемент, кроме элемента с номером 0, вляется **входом** ровно для одного порогового элемента. А именно, для каждого i, такого что $1 \le i \le N + M - 1$, элемент с номером i является входом для элемента с номером $P[i]$, где $0 \le P[i] \le N - 1$. При этом выполняется неравенство $P[i] < i$. Будем для удобства считать, что $P[0] = -1$. Каждый пороговый элемент имеет один или больше входов. Стартовые элементы не имеют входов.

Каждый элемент может находиться в **состоянии** равном 0 или 1. Начальные состояния стартовых элементов задаются массивом A, содержащем M целых чисел. А именно, для каждого j, такого что $0 \le j \le M - 1$, начальное состояние стартового элемента с номером $N + j$ равно $A[j]$.

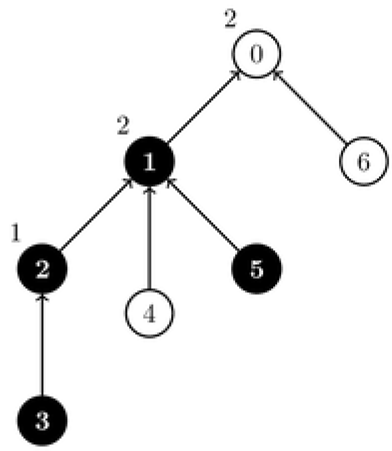
Состояние порогового элемента зависит от состояний его входов и определяется следующим образом. Сначала каждому пороговому элементу назначается его **параметр**. Параметр, назначаемый элементу с c входами должен быть целым числом в диапазоне от 1 до c (включительно). После этого состояние порогового элемента с параметром p равно 1, если хотя бы p его входов находятся в состоянии 1, иначе его состояние равно 0.

Например, пусть есть $N = 3$ пороговых элементов и $M = 4$ стартовых элементов. Входы для элемента 0 — это элементы 1 и 6, входы элемента 1 — это элементы 2, 4 и 5, а единственный вход элемента 2 — это элемент 3.

Описанная выше схема показана на следующем рисунке.



Пусть стартовые элементы с номерами 3 и 5 находятся в состоянии 1, а стартовые элементы с номерами 4 и 6 находятся в состоянии 0. Пусть мы назначили параметры 1, 2 и 2 пороговым элементам 2, 1 и 0, соответственно. В этом случае элемент с номером 2 находится в состоянии 1, элемент с номером 1 находится в состоянии 1, а элемент с номером 0 находится в состоянии 0. Результат назначения значений параметрам пороговых элементов и вычисления их состояний показаны на следующем рисунке. Элементы, находящиеся в состоянии 1, закрашены черным.



К состояниям, в которых находятся стартовые элементы, будут применены Q обновлений. Каждое обновление задается двумя целыми числами L и R ($N \le L \le R \le N + M - 1$) и изменяет состояние всех стартовых элементов с номерами от L до R, включительно, на противоположные. А именно, для каждого i, такого что $L \le i \le R$, стартовый элемент с номером i изменяет свое состояние на 1, если перед этим его состояние было равно 0, либо на 0, если его состояние было равно 1. Элемент после этого остается в новом состоянии, пока оно не будет изменено дальнейшими обновлениями.

Ваша задача — определить после каждого обновления, сколько существует способов назначить параметры пороговым элементам таким образом, чтобы элемент с номером 0 находился в состоянии 1. Два способа назначить параметры считаются различными, если существует хотя бы один пороговый элемент, которому в этих способах назначены различные параметры. Поскольку число способов может оказаться большим, необходимо вычислить его по модулю $1\ 000\ 002\ 022$.

Например, в приведенном выше примере существует 6 различных назначений параметров пороговым элементам, поскольку элементы с номерами 0, 1 и 2 имеют 2, 3 и 1 вход, соответственно. В 2 из этих 6 способов элемент с номером 0 находится в состоянии 1.

Implementation Details

Вам необходимо реализовать две функции.

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

- N: количество пороговых элементов.
- M: количество стартовых элементов.
- P: массив длиной $N + M$, задающий для каждого элемента, входом какого порогового элемента он является.
- A: массив длиной M, задающий начальные значения стартовых элементов.
- Эта функция будет вызывана ровно один раз, до всех вызовов функции `count_ways`.

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

- L, R: границы отрезка стартовых элементов, состояния которых изменяются на противоположные.
- Функция должна вернуть количество способов назначить параметры пороговым элементам после изменения состояния стартовых элементов запроса, таких что элемент с номером 0 находится в состоянии 1. Результат необходимо взять по модулю $1\ 000\ 002\ 022$.
- Эта функция будет вызвана ровно Q раз.

Example

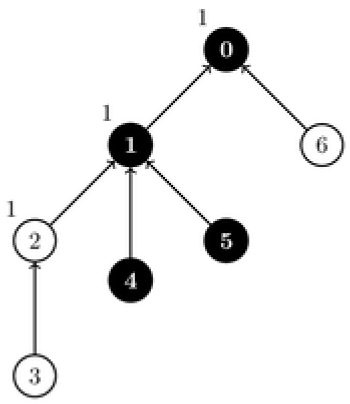
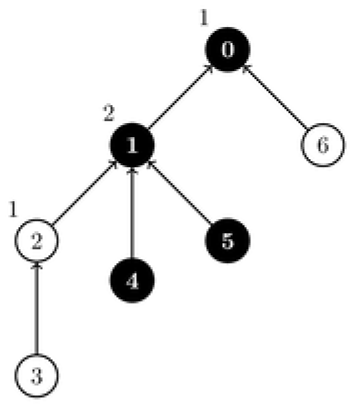
Рассмотрим следующую последовательность вызовов

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

Этот пример задает схему, показанную выше в условии.

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

Этот вызов изменяет состояние стартовых элементов с номерами 3 и 4, то есть состояние элемента 3 становится равно 0, а состояние элемента 4 становится равно 1. Два способа назначить параметры пороговым элементам, чтобы элемент с номером 0 был в состоянии 1, показаны на рисунке ниже.

Способ 1	**Способ** 2
	

Во всех других способах назначить параметры пороговым элементам элемент с номером 0 будет находиться в состоянии 0. Таким образом функция должна вернуть 2.

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

Этот вызов изменяет состояние стартовых элементов с номерами 4 и 5. В результате все стартовые элементы находятся в состоянии 0, и для любого способа назначить параметры пороговым элементам элемент с номером 0 будет находиться в состоянии 0. Таким образом, функция должна вернуть 0.

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

После этого вызова состояния всех стартовых элементов равны 1. В результате любое назначение параметров приводит к тому, что элемент с номером 0 находится в состоянии 1. Следовательно функция должна вернуть 6.

Constraints

- $1 \le N, M \le 100\ 000$
- $1 \le Q \le 100\ 000$
- $P[0] = -1$
- $0 \le P[i] < i$ и $P[i] \le N - 1$ (для всех i, таких что $1 \le i \le N + M - 1$)

- Каждый пороговый элемент имеет хотя бы один вход (для каждого i, такого что $0 \leq i \leq N-1$, существует индекс x, такой что $i < x \leq N+M-1$ и $P[x] = i$).
- $0 \leq A[j] \leq 1$ (для всех j, таких что $0 \leq j \leq M-1$)
- $N \leq L \leq R \leq N+M-1$

Subtasks

1. (2 балла) $N = 1$, $M \leq 1000$, $Q \leq 5$
2. (7 баллов) $N, M \leq 1000$, $Q \leq 5$, каждый пороговый элемент имеет ровно два входа.
3. (9 баллов) $N, M \leq 1000$, $Q \leq 5$
4. (4 балла) $M = N+1$, $M = 2^z$ (для некоторого положительного целого числа z), $P[i] = \lfloor \frac{i-1}{2} \rfloor$ (для каждого i, такого что $1 \leq i \leq N+M-1$), $L = R$
5. (12 баллов) $M = N+1$, $M = 2^z$ (для некоторого положительного целого числа z), $P[i] = \lfloor \frac{i-1}{2} \rfloor$ (для каждого i, такого что $1 \leq i \leq N+M-1$)
6. (27 баллов) Каждый пороговый элемент имеет ровно два входа.
7. (28 баллов) $N, M \leq 5000$
8. (11 баллов) Нет дополнительных ограничений.

Sample Grader

Пример грейдера читает входные данные в следующем формате:

- строка 1: $N \ M \ Q$
- строка 2: $P[0] \ P[1] \ \ldots \ P[N+M-1]$
- строка 3: $A[0] \ A[1] \ \ldots \ A[M-1]$
- строка $4+k$ ($0 \leq k \leq Q-1$): $L \ R$ для обновления k

Пример грейдера выводит результат в следующем формате:

- строка $1+k$ ($0 \leq k \leq Q-1$): возвращаемое значение `count_ways` для обновления k