디지털 회로

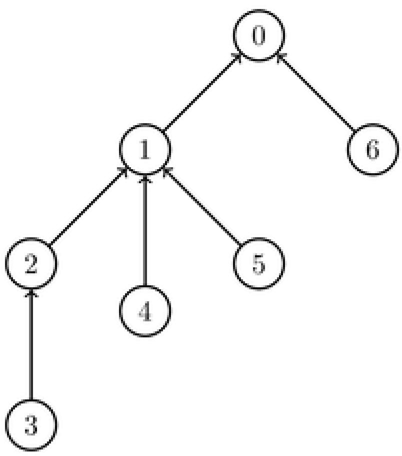
$N+M$개의 **게이트**로 구성된 회로가 있다. 게이트들은 0부터 $N+M-1$까지 번호가 붙어 있다. 게이트 0부터 $N-1$까지는 **임계 게이트**들이고, 게이트 N부터 $N+M-1$까지는 **소스 게이트**들이다.

게이트 0을 제외한 각 게이트의 출력은 하나이고 정확히 하나의 임계 게이트의 **입력**으로 연결된다. 좀더 구체적으로, 각 게이트 i의 $(1 \le i \le N+M-1)$ 출력은 게이트 $P[i]$의 입력이다. $(0 \le P[i] \le N-1)$ 중요하게, $P[i] < i$가 항상 성립한다. 또, $P[0] = -1$이다. 즉, 게이트 0의 출력은 다른 어떤 게이트의 입력으로도 연결되지 않는다. 모든 임계 게이트는 하나 이상의 입력을 가진다. 모든 소스 게이트는 입력이 없다.

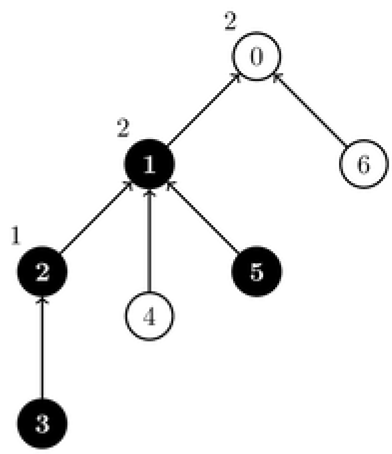
각 게이트는 0 혹은 1의 값이 될 수 있는 **상태**를 가진다. 게이트의 출력은 그 게이트의 상태와 같다. 소스 게이트들의 상태는 입력 배열 A로 (배열 크기 M) 주어진다. 즉, 각 j에 대해 $(0 \le j \le M-1)$, $A[j]$의 값이 게이트 $N+j$의 상태이다.

각 임계 게이트의 상태는 해당 게이트의 입력에 따라 다음과 같이 결정된다. 각 임계 게이트에는 임계치를 결정하는 **파라미터**가 정해져 있다. c개의 입력을 가진 임계 게이트의 파라미터는 1 이상 c 이하의 정수이다. 파라미터가 p인 임계 게이트의 상태는, 입력 중 p개 이상이 1인 경우 1이고, 그렇지 않은 경우 0이다.

예를 들어, 아래 그림처럼 $N=3$개의 임계 게이트와 $M=4$개의 소스 게이트가 있는 회로가 있다고 하자. 게이트 0의 입력은 게이트 1과 6(의 출력)이고, 게이트 1의 입력은 게이트 $2, 4, 5$이며, 게이트 2의 입력은 게이트 3이다.



이 회로에서 게이트 3과 5의 상태는 1이고 게이트 4와 6의 상태는 0이라고 하자. 게이트 $2, 1, 0$의 파라미터는 각각 $1, 2, 2$라고 하자. 이 경우, 게이트 2의 상태는 1, 게이트 1의 상태는 1, 게이트 0의 상태는 0이 된다. 위의 상황은 아래 그림에 표시되어 있다. 게이트의 상태가 1인 것들이 검은 색으로 표시되어 있다.



소스 게이트들의 상태는 Q번 업데이트된다. 각 업데이트는 두 정수 L, R로 $(N \le L \le R \le N+M-1)$ 표현된다. 업데이트의 의미는 번호가 L부터 R까지인 모든 소스 게이트의 상태를 뒤집는 것이다. 뒤집는다는 것의 의미는 0을 1로, 1을 0으로 바꾸는 것이다. 업데이트에 의해 변경된 게이트들의 상태는 이후의 업데이트에 영향을 받지 않는 한 유지된다.

초기 상태를 입력 받고, 각 업데이트 이후에 게이트 0의 상태가 1이 되도록 만들 수 있는 임계 게이트 파라미터 설정 방법의 경우의 수를 계산하는 프로그램을 작성하라. 두 파라미터 설정 방법이 다르다는 것은, 임계 게이트 중 하나라도 다른 파라미터 값을 가진다는 것으로 정의된다. 경우의 수 값이 매우 클 수 있으므로 그 값을 $1\ 000\ 002\ 022$으로 나눈 나머지를 결과로 제시해야 한다.

위의 예에서 게이트 $0, 1, 2$는 각각 $2, 3, 1$개의 입력이 있으므로, 가능한 파라미터 설정 방법은 6가지가 있다. 가능한 방법들 중 2가지에서 게이트 0의 상태는 1이 된다.

Implementation Details

다음 2개의 함수를 구현해야 한다.

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

- N: 임계 게이트의 개수.
- M: 소스 게이트의 개수.
- P: 임계 게이트의 입력을 표현한 크기 $N+M$인 배열.
- A: 소스 게이트의 초기 상태를 표현한 크기 M인 배열.
- 이 함수는 모든 count_ways 호출 이전에 정확히 한번 호출된다.

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

- L, R: 업데이트에 의해 뒤집히는 소스 게이트들의 번호 범위.
- 이 함수는 지정된 업데이트를 수행한 후, 게이트 0의 상태가 1이 되게 만드는 파라미터 설정 방법의 경우의 수를 $1\ 000\ 002\ 022$로 나눈 나머지를 리턴해야 한다.
- 이 함수는 정확히 Q번 호출된다.

Example

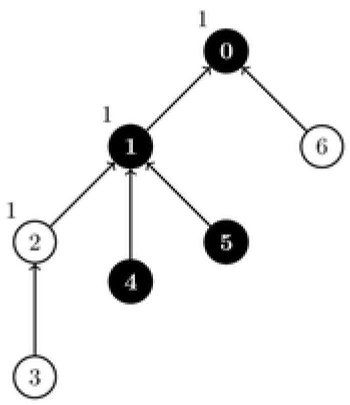
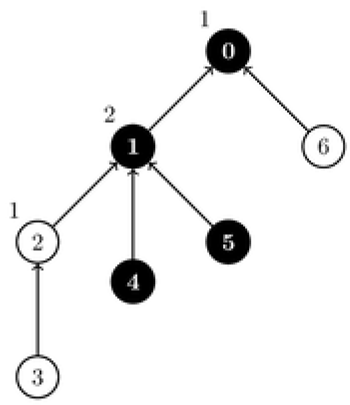
다음 호출들을 보자:

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

이 회로는 본문에서 설명한 것과 같다.

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

게이트 3과 4를 뒤집는다. 즉, 게이트 3의 상태는 0이 되고 게이트 4의 상태는 1이 된다. 게이트 0의 상태가 1이 되게 하는 2가지 파라미터 설정이 아래 그림에 표현되어 있다.

설정 1	설정 2
	

위의 방법 이외의 모든 다른 설정에서 게이트 0의 상태가 0이 된다. 따라서, 이 함수 호출은 2를 리턴해야 한다.

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

게이트 4와 5를 뒤집는다. 모든 소스 게이트의 상태가 0이 되었다. 이 경우 파라미터 설정을 어떻게 해도 게이트 0의 상태는 0이다. 따라서, 이 함수 호출은 0을 리턴해야 한다.

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

모든 소스 게이트의 상태가 1로 바뀐다. 이 경우 파라미터 설정을 어떻게 해도 게이트 0의 상태는 1이다. 따라서, 이 함수 호출은 6을 리턴해야 한다.

Constraints

- $1 \le N, M \le 100\ 000$

- $1 \le Q \le 100\,000$
- $P[0] = -1$
- $0 \le P[i] < i$이고 $P[i] \le N-1$ ($1 \le i \le N+M-1$)
- 각 임계 게이트는 최소 하나의 입력이 있다. (모든 $i(0 \le i \le N-1)$에 대해 최소 하나의 $x(i < x \le N+M-1)$가 있어서 $P[x] = i$이다.)
- $0 \le A[j] \le 1$ ($0 \le j \le M-1$)
- $N \le L \le R \le N+M-1$

Subtasks

1. (2 points) $N = 1$, $M \le 1000$, $Q \le 5$
2. (7 points) $N, M \le 1000$, $Q \le 5$, 각 임계 게이트에는 정확히 2개의 입력이 있다.
3. (9 points) $N, M \le 1000$, $Q \le 5$
4. (4 points) $M = N+1$, $M = 2^z$ (양의 정수 z), $P[i] = \lfloor \frac{i-1}{2} \rfloor$ ($1 \le i \le N+M-1$), $L = R$
5. (12 points) $M = N+1$, $M = 2^z$ (양의 정수 z), $P[i] = \lfloor \frac{i-1}{2} \rfloor$ ($1 \le i \le N+M-1$)
6. (27 points) 각 임계 게이트에는 정확히 2개의 입력이 있다.
7. (28 points) $N, M \le 5000$
8. (11 points) 추가적인 제한이 없다.

Sample Grader

샘플 그레이더는 다음의 양식으로 입력을 받는다:

- line 1: $N\ M\ Q$
- line 2: $P[0]\ P[1]\ \ldots\ P[N+M-1]$
- line 3: $A[0]\ A[1]\ \ldots\ A[M-1]$
- line $4+k$ ($0 \le k \le Q-1$): 업데이트k의 L과 R

샘플 그레이더는 다음의 출력을 생성한다:

- line $1+k$ ($0 \le k \le Q-1$): 업데이트 k에 대한 `count_ways`의 리턴 값