

## 입자 가속기 (particles)

KOI 연구소는 입자 가속기를 이용하여 연구를 진행하고 있다. KOI 연구소의 입자 가속기는  $N$ 개의 방과 방들을 잇는  $N - 1$ 개의 양방향 통로로 이루어져 있으며, 임의의 서로 다른 두 방들을 통로만을 사용하여 오갈 수 있다. 즉, KOI 연구소의 입자 가속기는 트리 구조를 이룬다.

입자 가속기의 방들에는 0부터  $N - 1$ 까지의 서로 다른 번호가 붙어 있으며, 입자 가속기의 통로들에는 0부터  $N - 2$ 까지의 서로 다른 번호가 붙어 있다. 모든  $0 \leq i \leq N - 2$ 에 대해  $i$ 번 통로는  $A[i]$ 번 방과  $B[i]$ 번 방을 연결한다.

최근 KOI 연구소는 IOI 입자 충돌 실험을 진행하고 있다. IOI 입자는 생성하는 것도 매우 어렵기 때문에, 입자 가속기의 방마다 최대 1개씩만 생성을 시도할 수 있다. 어떤 방에서 IOI 입자를 생성하는 데 성공하면 IOI 입자 1개가 그 방에 정지한 상태로 존재하게 된다. 반대로, IOI 입자를 생성하는 데 실패하면 실험 장비 점검을 위해 그 방은 폐쇄되어 사용할 수 없게 된다.

KOI 연구소는 몇 개의 방을 골라 IOI 입자 생성을 시도한 후 실험을 진행하고자 한다. 실험은 여러 번의 충돌 실험이 순차적으로 진행되는 형태이며, 각 충돌 실험에서는 IOI 입자가 있는 방 2개를 골라 IOI 입자 충돌 실험을 진행한다. 이때, 선택된 방을 연결하는 경로 위에 IOI 입자가 있는 방이 있어서는 안 되며, IOI 입자 생성에 실패한 방이 있어서는 안 된다. IOI 입자 충돌 실험을 진행하고 나면 충돌 실험에 사용된 두 입자는 소멸한다. IOI 입자 생성에 성공했지만 IOI 입자가 더 이상 존재하지 않는 방의 경우, 추후 진행되는 충돌 실험에서 선택된 2개의 방을 연결하는 경로 위에 존재해도 된다는 점에 유의하라.

여러분은 KOI 연구소가 충돌 실험을 최대 몇 회 진행할 수 있는지 구해야 한다. 초기에는 입자 가속기의 모든 방에 IOI 입자가 없으며, 모든 방에 대해 IOI 입자 생성을 시도할 수 있다. KOI 연구소는 총  $Q$ 번 IOI 입자 생성을 시도하며, 여러분은 각 생성 시도마다 현재 상태에서 실험을 진행할 경우 충돌 실험을 최대 몇 회 진행할 수 있는지 구해야 한다.

## 함수 목록 및 정의

여러분은 아래 함수들을 구현해야 한다.

```
void initialize(int N, std::vector<int> A, std::vector<int> B)
```

- $N$ : 입자 가속기의 방 개수.
- $A, B$ : 크기가  $N - 1$ 인 정수 배열. 모든 정수  $0 \leq i \leq N - 2$ 에 대해,  $A[i]$ 번 방과  $B[i]$ 번 방을 잇는 통로가 존재한다.
- 이 함수는 초기에 단 한 번만 호출된다.

```
int generate(int v, bool result)
```

- 이 함수는  $v$ 번 방에서 진행된 IOI 입자 생성 시도를 나타낸다.
- 이 함수는 `initialize` 함수가 호출되고 나서 총  $Q$ 번 호출된다.
- $v$ : IOI 입자 생성을 시도한 방의 번호. 이 함수가 호출되기 이전에  $v$ 번 방에서 IOI 입자 생성을 시도하지 않았음이 보장된다.
- `result`: IOI 입자 생성 여부. 값이 `true`일 경우, IOI 입자 생성에 성공했다는 뜻이며,  $v$ 번 방에 IOI 입자가 존재하게 된다. 값이 `false`일 경우, IOI 입자 생성에 실패했다는 뜻이며,  $v$ 번 방은 폐쇄되어 사용할 수 없게 된다.

- 이 함수는 현재 상태에서 실험을 진행할 경우 할 수 있는 충돌 실험의 최대 횟수를 반환해야 한다.

## 제약 조건

- $2 \leq Q \leq N \leq 200\,000$
- KOI 연구소의 입자 가속기는 트리 구조를 이룬다.
- 모든  $i$ 에 대해  $0 \leq A[i], B[i] \leq N - 1; A[i] \neq B[i]$  ( $0 \leq i \leq N - 2$ )
- 모든 `generate` 함수 호출에 대해  $0 \leq v \leq N - 1; v$ 는 서로 다른 값이다.

## 부분문제

1. (9점)
  - $2 \leq Q \leq N \leq 5\,000$
2. (16점)
  - 모든  $i$ 에 대해  $A[i] = i; B[i] = i + 1$  ( $0 \leq i \leq N - 2$ )
3. (20점)
  - `result = false`인 `generate` 함수 호출은 최대 20회이다.
4. (23점)
  - `result = true`인 `generate` 함수 호출은 최대 20회이다.
5. (32점)
  - 추가적인 제약 조건이 없다.

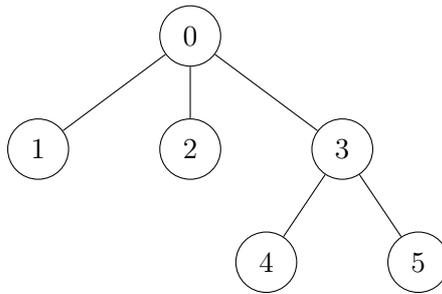
**예제 1**

$N = 6$ ,  $A = [0, 0, 0, 3, 3]$ ,  $B = [1, 2, 3, 4, 5]$ 인 경우를 생각해 보자.

그레이더는 다음 함수를 호출한다.

```
initialize(6, [0, 0, 0, 3, 3], [1, 2, 3, 4, 5])
```

아래 그림은 KOI 연구소의 입자 가속기 구조를 나타낸다.

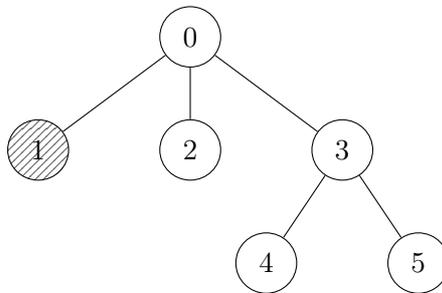


1번 방에서 IOI 입자 생성에 성공한 경우를 생각해 보자.

그레이더는 다음 함수를 호출한다.

```
generate(1, true)
```

아래 그림은 입자 가속기의 현재 상태를 나타낸다. 현재 1번 방에는 IOI 입자가 존재하는 상태이며, 사선 무늬로 표시되어 있다. 충돌 실험을 진행할 수 없으므로 함수는 0을 반환해야 한다.

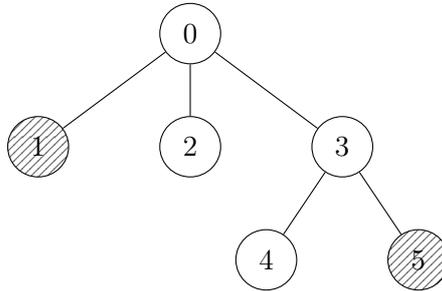


5번 방에서 IOI 입자 생성에 성공한 경우를 생각해 보자.

그레이더는 다음 함수를 호출한다.

```
generate(5, true)
```

아래 그림은 입자 가속기의 현재 상태를 나타낸다. 현재 1, 5번 방에는 IOI 입자가 존재하는 상태이며, 사선 무늬로 표시되어 있다. 1번 방과 5번 방을 골라 충돌 실험을 진행할 수 있다. 충돌 실험을 2회 이상 진행할 수 없으므로 함수는 1을 반환해야 한다.

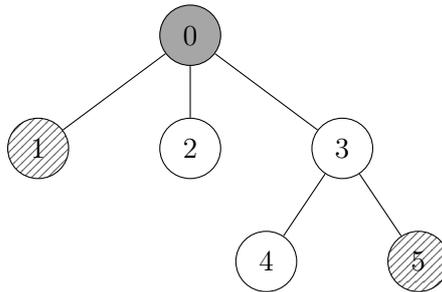


0번 방에서 IOI 입자 생성에 실패한 경우를 생각해 보자.

그레이더는 다음 함수를 호출한다.

```
generate(0, false)
```

아래 그림은 입자 가속기의 현재 상태를 나타낸다. 현재 1, 5번 방에는 IOI 입자가 존재하는 상태이며, 사선 무늬로 표시되어 있다. 현재 0번 방은 폐쇄된 상태이며, 회색으로 표시되어 있다. 1번 방과 5번 방을 연결하는 경로 위에 0번 방이 있기 때문에 충돌 실험을 진행할 수 없다. 함수는 0을 반환해야 한다.

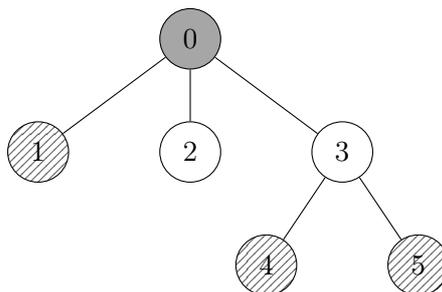


4번 방에서 IOI 입자 생성에 성공한 경우를 생각해 보자.

그레이더는 다음 함수를 호출한다.

```
generate(4, true)
```

아래 그림은 입자 가속기의 현재 상태를 나타낸다. 현재 1, 4, 5번 방에는 IOI 입자가 존재하는 상태이며, 사선 무늬로 표시되어 있다. 현재 0번 방은 폐쇄된 상태이며, 회색으로 표시되어 있다. 4번 방과 5번 방을 골라 충돌 실험을 진행할 수 있다. 충돌 실험을 2회 이상 진행할 수 없으므로 함수는 1을 반환해야 한다.

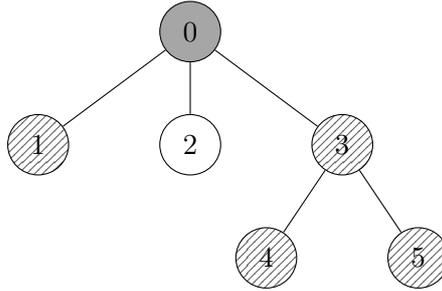


3번 방에서 IOI 입자 생성에 성공한 경우를 생각해 보자.

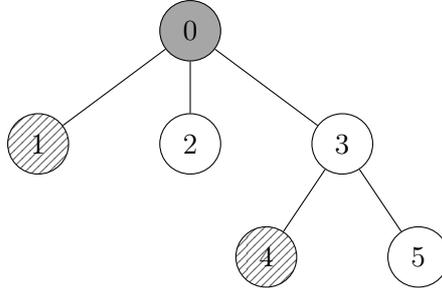
그레이더는 다음 함수를 호출한다.

```
generate(3, true)
```

아래 그림은 입자 가속기의 현재 상태를 나타낸다. 현재 1, 3, 4, 5번 방에는 IOI 입자가 존재하는 상태이며, 사선 무늬로 표시되어 있다. 현재 0번 방은 폐쇄된 상태이며, 회색으로 표시되어 있다. 4번 방과 5번 방을 연결하는 경로 위에 3번 방이 있기 때문에 4번 방과 5번 방을 골라 충돌 실험을 진행할 수 없다. 그러나, 3번 방과 5번 방을 골라 충돌 실험을 진행할 수 있다.



아래 그림은 3번 방과 5번 방을 골라 충돌 실험을 진행하고 난 입자 가속기의 상태를 나타낸다. 1번 방과 4번 방을 연결하는 경로 위에 0번 방이 있기 때문에 충돌 실험을 진행할 수 없다. 어떤 방법으로도 충돌 실험을 2회 이상 진행할 수 없으므로 함수는 1을 반환해야 한다.



## Sample grader

Sample grader는 아래와 같은 형식으로 입력을 받는다.

- Line 1:  $N Q$
- Line  $2 + i$  ( $0 \leq i \leq N - 2$ ):  $A[i] B[i]$
- Line  $N + 1 + j$  ( $0 \leq j \leq Q - 1$ ):  $v result$  ( $result$ 가 `true`면 1, `false`면 0을 입력한다.)

Sample grader는 다음을 출력한다.

- Line  $1 + j$  ( $0 \leq j \leq Q - 1$ ):  $j$ 번째 `generate` 함수가 반환한 값

Sample grader는 실제 채점에서 사용하는 그레이더와 다를 수 있음에 유의하라.