



Digital Circuit

Có một mạch điện, bao gồm $N + M$ cổng được đánh số từ 0 đến $N + M - 1$. Các cổng từ 0 đến $N - 1$ là các **cổng ngưỡng**, trong khi các cổng từ N đến $N + M - 1$ là các **cổng nguồn**.

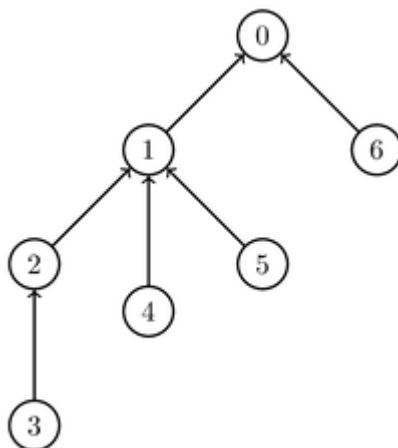
Mỗi cổng, ngoại trừ cổng 0, là một **đầu vào** cho đúng một cổng ngưỡng. Cụ thể, với mỗi i sao cho $1 \leq i \leq N + M - 1$, cổng i là một đầu vào cho cổng $P[i]$, với $0 \leq P[i] \leq N - 1$. Lưu ý là ta còn có $P[i] < i$. Thêm nữa, giả thiết $P[0] = -1$. Mỗi cổng ngưỡng có một hoặc nhiều đầu vào. Các cổng nguồn không có bất kỳ đầu vào nào.

Mỗi cổng có một **trạng thái** là 0 hoặc 1. Trạng thái ban đầu của các cổng nguồn được cho bởi mảng A gồm M số nguyên. Nghĩa là, với mỗi j sao cho $0 \leq j \leq M - 1$, trạng thái ban đầu của cổng nguồn $N + j$ là $A[j]$.

Trạng thái của mỗi cổng ngưỡng phụ thuộc vào trạng thái các đầu vào của nó và được xác định như sau. Đầu tiên, mỗi cổng ngưỡng được gán một **tham số** ngưỡng. Tham số được gán cho cổng ngưỡng có c đầu vào phải là một số nguyên trong khoảng từ 1 đến c (bao gồm cả hai đầu mút). Khi đó, trạng thái của cổng ngưỡng với tham số p là 1, nếu ít nhất p đầu vào trong các đầu vào của nó có trạng thái 1 và có trạng thái 0 nếu ngược lại.

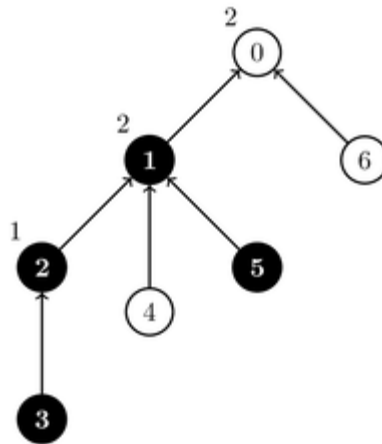
Ví dụ, giả sử có $N = 3$ cổng ngưỡng và $M = 4$ cổng nguồn. Các đầu vào cho cổng 0 là các cổng 1 và 6, các đầu vào cho cổng 1 là các cổng 2, 4 và 5, và đầu vào duy nhất cho cổng 2 là cổng 3.

Ví dụ này được minh họa trong hình sau.



Giả sử các cổng nguồn 3 và 5 có trạng thái 1, trong khi các cổng nguồn 4 và 6 có trạng thái 0. Giả sử ta gán các tham số 1, 2 và 2 tương ứng cho các cổng ngưỡng 2, 1 và 0. Trong trường hợp này,

cổng 2 có trạng thái 1, cổng 1 có trạng thái 1 và cổng 0 có trạng thái 0. Việc gán các giá trị tham số và trạng thái này được minh họa trong hình sau. Các cổng có trạng thái là 1 được đánh dấu bởi màu đen.



Trạng thái của các cổng nguồn sẽ có Q lần cập nhật. Mỗi lần cập nhật được mô tả bởi hai số nguyên L và R ($N \leq L \leq R \leq N + M - 1$) và chuyển đổi trạng thái của tất cả các cổng nguồn được đánh số từ L đến R , bao gồm cả hai đầu mút. Nghĩa là, với mỗi i sao cho $L \leq i \leq R$, cổng nguồn i thay đổi trạng thái của nó thành 1, nếu trạng thái của nó là 0 hoặc thành 0, nếu trạng thái của nó là 1. Trạng thái mới của mỗi cổng được chuyển đổi không thay đổi cho đến khi nó có thể được chuyển đổi bởi một trong các lần cập nhật sau đó.

Nhiệm vụ của bạn là sau mỗi lần cập nhật, đếm xem có bao nhiêu cách gán tham số khác nhau cho các cổng ngưỡng để cổng 0 có trạng thái 1. Hai cách gán được coi là khác nhau nếu tồn tại ít nhất một cổng ngưỡng mà tham số của nó có giá trị khác nhau trong cả hai cách gán. Vì số lượng cách có thể lớn, bạn cần tính phần dư trong phép chia cho 1 000 002 022.

Lưu ý trong ví dụ trên, có 6 cách gán tham số khác nhau cho các cổng ngưỡng, vì các cổng 0, 1 và 2 có các đầu vào tương ứng là 2, 3 và 1. Ở 2 trong số 6 cách gán này, cổng 0 có trạng thái 1.

Chi tiết cài đặt

Nhiệm vụ của bạn là cài đặt hai hàm sau.

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

- N : số lượng các cổng ngưỡng.
- M : số lượng các cổng nguồn.
- P : mảng $N + M$ phần tử mô tả các đầu vào cho các cổng ngưỡng.
- A : mảng M phần tử mô tả các trạng thái ban đầu của các cổng nguồn.
- Hàm này được gọi đúng một lần, trước bất kỳ lời gọi hàm `count_ways` nào.

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

- L, R : phạm vi chỉ số các cổng nguồn mà có các trạng thái được chuyển đổi.
- Hàm này trước tiên cần thực hiện các lần cập nhật, và tiếp theo trả về phần dư của số cách gán các tham số cho các cổng ngưỡng trong phép chia cho 1 000 002 022, dẫn đến cổng 0 có trạng thái 1.
- Hàm này được gọi đúng Q lần.

Ví dụ

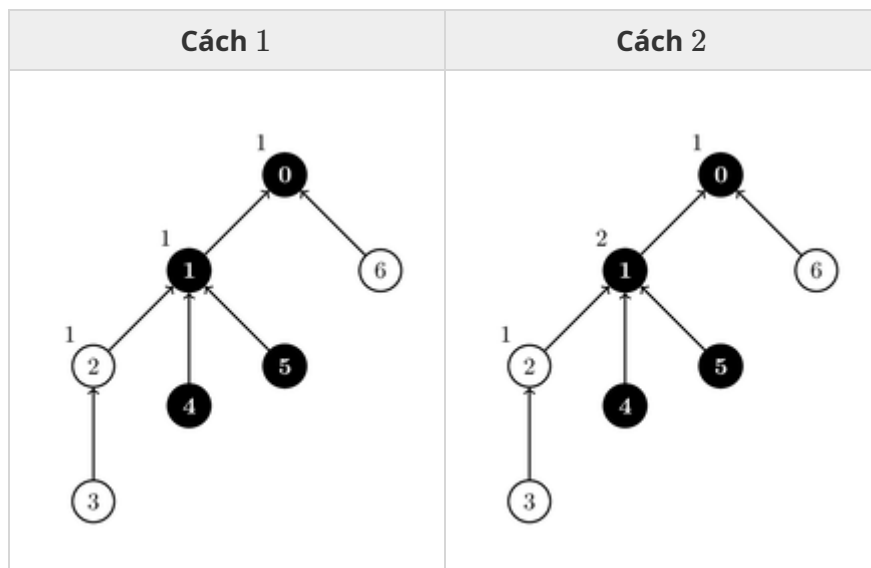
Xét chuỗi các lời gọi sau:

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

Ví dụ này được minh họa trong phần mô tả bài toán ở trên.

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

Hàm này chuyển đổi trạng thái của cổng 3 và 4, tức là trạng thái của cổng 3 trở thành 0 và trạng thái của cổng 4 trở thành 1. Hai cách gán các tham số dẫn đến cổng 0 có trạng thái 1 được minh họa trong các hình sau.



Trong tất cả các cách gán tham số khác, cổng 0 có trạng thái 0. Vì vậy, hàm cần trả về 2.

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

Hàm này chuyển đổi trạng thái của các cổng 4 và 5. Kết quả là, tất cả các cổng nguồn đều có trạng thái 0 và đối với bất kỳ cách gán tham số nào, cổng 0 có trạng thái 0. Vì vậy, hàm cần trả về 0.

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

Hàm này thay đổi trạng thái của tất cả các cổng nguồn thành 1. Kết quả là, đối với bất kỳ cách gán tham số nào, cổng 0 có trạng thái 1. Vì vậy, hàm cần trả về 6.

Ràng buộc

- $1 \leq N, M \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $P[0] = -1$
- $0 \leq P[i] < i$ và $P[i] \leq N - 1$ (với mỗi i sao cho $1 \leq i \leq N + M - 1$)
- Mỗi cổng ngưỡng có ít nhất một đầu vào (với mỗi i sao cho $0 \leq i \leq N - 1$ tồn tại một chỉ số x sao cho $i < x \leq N + M - 1$ và $P[x] = i$).
- $0 \leq A[j] \leq 1$ (với mỗi j sao cho $0 \leq j \leq M - 1$)
- $N \leq L \leq R \leq N + M - 1$

Subtask

1. (2 điểm) $N = 1, M \leq 1000, Q \leq 5$
2. (7 điểm) $N, M \leq 1000, Q \leq 5$, mỗi cổng ngưỡng có đúng hai đầu vào.
3. (9 điểm) $N, M \leq 1000, Q \leq 5$
4. (4 điểm) $M = N + 1, M = 2^z$ (với một số nguyên dương z nào đó), $P[i] = \lfloor \frac{i-1}{2} \rfloor$ (với mỗi i sao cho $1 \leq i \leq N + M - 1$), $L = R$
5. (12 điểm) $M = N + 1, M = 2^z$ (với một số nguyên dương z nào đó), $P[i] = \lfloor \frac{i-1}{2} \rfloor$ (với mỗi i sao cho $1 \leq i \leq N + M - 1$)
6. (27 điểm) Mỗi cổng ngưỡng có đúng hai đầu vào.
7. (28 điểm) $N, M \leq 5000$
8. (11 điểm) Không có ràng buộc gì thêm.

Trình chấm mẫu

Trình chấm mẫu đọc đầu vào theo định dạng sau:

- dòng 1: $N M Q$
- dòng 2: $P[0] P[1] \dots P[N + M - 1]$
- dòng 3: $A[0] A[1] \dots A[M - 1]$
- dòng $4 + k$ ($0 \leq k \leq Q - 1$): $L R$ cho lần cập nhật thứ k

Trình chấm mẫu in kết quả của bạn theo định dạng sau:

- dòng $1 + k$ ($0 \leq k \leq Q - 1$): giá trị trả về của `count_ways` cho lần cập nhật thứ k