



الدارة الرقمية

لدينا دائرة مؤلفة من $N + M$ بوابة مرقمة من 0 إلى $N + M - 1$. تدعى البوابات المرقمة من 0 إلى $N - 1$ بوابات العتبة، بينما تدعى البوابات المرقمة من N إلى $N + M - 1$ بوابات المصدر.

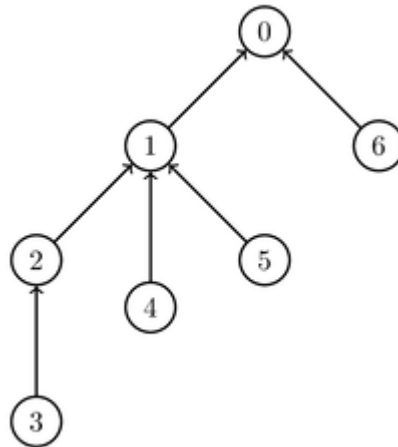
كل بوابة ما عدا البوابة رقم 0 تشكل دخلاً لبوابة عتبة واحدة تماماً. أي أنه لكل i حيث $1 \leq i \leq N + M - 1$ ، البوابة رقم i هي دخل لبوابة $P[i]$ حيث $0 \leq P[i] \leq N - 1$. من المهم أن تعلم أيضاً أن $P[i] < i$. بالإضافة إلى ذلك، نفترض أن $P[0] = -1$. لكل بوابة عتبة دخل واحد أو أكثر. لا تملك بوابات المصدر أية قيم دخل.

لكل بوابة حالة إما 0 أو 1. تكون الحالات الابتدائية لبوابات المصدر معطاة بمصفوفة A من M عدداً صحيحاً. هذا يعني أنه من أجل كل j حيث $0 \leq j \leq M - 1$ ، الحالة الابتدائية لبوابة المصدر ذات الرقم $N + j$ تكون $A[j]$.

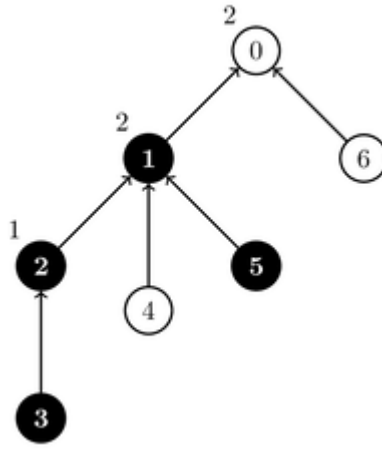
حالة كل بوابة عتبة تعتمد على حالات بوابات الدخل المتصلة بها ويتم تحديدها بالطريقة التالية. أولاً، يتم تعيين معامل عتبة لكل بوابة عتبة. المعامل الذي تم تعيينه لبوابة عتبة لها c دخل يجب أن يكون عدداً صحيحاً بين 1 و c (ضمناً). ثم تكون حالة بوابة العتبة التي تملك معاملًا ذو قيمة p مساوية ل 1 إذا وجد p دخلاً لها على الأقل لها حالة 1، وإلا تكون حالتها 0.

على سبيل المثال، لنفرض أن $N = 3$ عدد بوابات العتبة و $M = 4$ عدد بوابات المصدر. دخلا البوابة ذات الرقم 0 هما البوابتان 1 و 6، دخل البوابة رقم 1 هو البوابات 2 و 4 و 5، والدخل الوحيد للبوابة رقم 2 هو البوابة 3.

الشكل التالي يوضح المثال السابق.



لنفرض أن بوابات المصدر ذات الرقم 3 و 5 تملك الحالة 1، بينما تملك بوابات المصدر 4 و 6 الحالة 0. لنفرض أننا قمنا بتعيين قيم المعاملات 1 و 2 و 2 لبوابات العتبة 2 و 1 و 0 على الترتيب. في هذه الحالة تكون حالة البوابة 2 مساوية ل 1، وحالة البوابة 1 مساوية ل 1 وحالة البوابة 0 مساوية ل 0. في الشكل التالي يمكنك ملاحظة تعيين المعاملات وقيمة الحالات للبوابات. البوابات ذات الحالة 1 ملونة بالأسود.



سيتم إجراء Q تحديثاً على حالات بوابات المصدر. يوصف كل تحديث بعددين صحيحين L و R حيث $(N \leq L \leq R \leq N + M - 1)$ ويقوم بعكس حالة كل بوابات المصدر المرقمة بين L و R ضمناً. هذا يعني أنه من أجل كل i حيث $L \leq i \leq R$ ، فإن بوابة المصدر ذات الرقم i تبديل حالتها إلى 1 إذا كانت حالتها 0، أو تبديل حالتها إلى 0 إذا كانت حالتها 1. بعد كل تحديث، البوابات التي تم عكس حالتها تحافظ على قيم حالاتها الجديدة إلى أن يتم عكسها بأحد التحديثات اللاحقة ان وجدت.

هدفك هو أن تقوم بعدد التعيينات المختلفة لمعاملات بوابات العتبة التي تؤدي إلى أن تكون البوابة ذات الرقم 0 لها الحالة 1، وذلك بعد كل تحديث. يتم اعتبار تعيينين للمعاملات على أنهما مختلفين إذا وجدت بوابة عتبة واحدة على الأقل بقيمتين مختلفتين لمعاملها في كلا التعيينين. بما أن عدد التعيينات يمكن أن يكون كبيراً، عليك حساب باقي قسمة عدد التعيينات على العدد 1 000 002 022.

لاحظ أنه في المثال أعلاه، يوجد 6 تعيينات مختلفة لمعاملات بوابات العتبة، بما أن البوابات 0 و 1 و 2 لها 2 و 3 و 1 قيمة دخل على الترتيب. يوجد تعيينان 2 من التعيينات ال 6 تكون فيها حالة البوابة رقم 0 مساوية ل 1.

تفاصيل التحقيق

عليك أن تحقق إجرائيتين.

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

- N : عدد بوابات العتبة.
- M : عدد بوابات المصدر.
- P : مصفوفة بطول $N + M$ تصف دخل بوابات العتبة.
- A : مصفوفة بطول M تصف الحالة الابتدائية لبوابات المصدر.
- يتم استدعاء هذه الاجرائية لمرة واحدة فقط، وذلك قبل أي استدعاء ل `count_ways`.

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

- L, R : حدود مجال بوابات المصدر التي سيتم عكس حالاتها.
- يقوم هذا التابع بتطبيق التحديث المحدد بالدخل، ثم يعيد عدد الطرق، بعد أخذ باقي قسمته على 1 000 002 022، لتعيين معاملات بوابات العتبة بما يؤدي لأن تكون حالة البوابة رقم 0 مساوية ل 1.

- يتم استدعاء هذا التابع Q مرة تماماً.

مثال

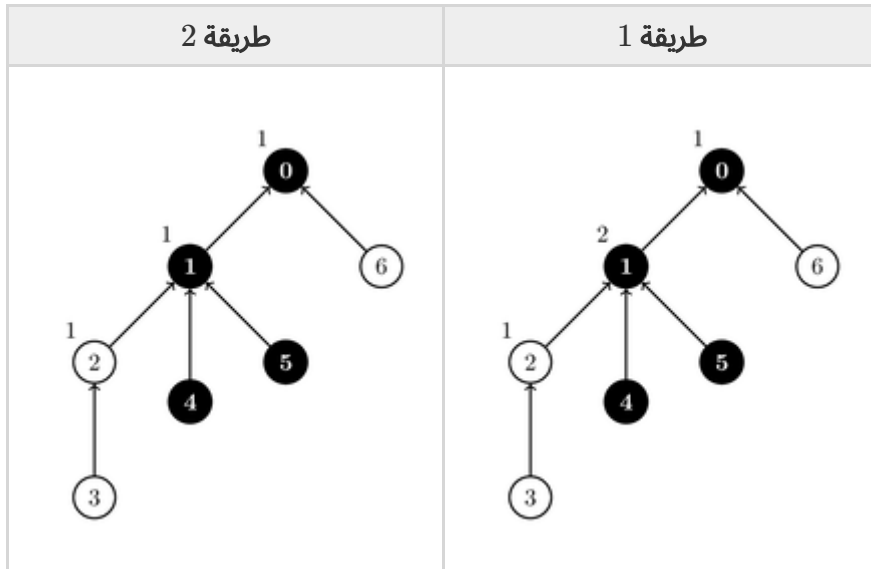
لتكن لدينا سلسلة الاستدعاءات التالية:

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

تم توضيح هذا الاستدعاء في المثال المرسوم أعلاه.

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

يقوم هذا الاستدعاء بعكس حالات البوابات 3 و 4، أي أن حالة البوابة 3 تصبح 0، وحالة البوابة 4 تصبح 1. يوجد طريقتين لتعيين قيم المعاملات بما يجعل حالة البوابة 0 مساوية ل 1، تم توضيحهما بالأشكال التالية.



بكل طرق التعيين الأخرى للمعاملات تكون حالة البوابة 0 مساوية ل 0. بالتالي يجب أن يرد التابع القيمة 2.

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

يقوم هذا الاستدعاء بعكس قيم حالات البوابات 4 و 5. نتيجة لذلك، تصبح قيم جميع بوابات المصدر مساوية ل 0، وبالتالي من أجل أي تعيين لقيم المعاملات ستكون حالة البوابة 0 مساوية ل 0. بالتالي يجب أن يرد التابع القيمة 0.

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

يقوم هذا الاستدعاء بتحويل حالة كل بوابات المصدر إلى 1. نتيجة لذلك، من أجل أي تعيين لقيم المعاملات ستكون حالة البوابة 0 مساوية ل 1. بالتالي يجب أن يرد التابع القيمة 6.

Constraints

- $1 \leq N, M \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $P[0] = -1$
- $0 \leq P[i] < i$ and $P[i] \leq N - 1$ (for each i such that $1 \leq i \leq N + M - 1$)
- Each threshold gate has at least one input (for each i such that $0 \leq i \leq N - 1$ there exists an index x such that $i < x \leq N + M - 1$ and $P[x] = i$).
- $0 \leq A[j] \leq 1$ (for each j such that $0 \leq j \leq M - 1$)
- $N \leq L \leq R \leq N + M - 1$

Subtasks

1. (2 points) $N = 1, M \leq 1000, Q \leq 5$
2. (7 points) $N, M \leq 1000, Q \leq 5$, each threshold gate has exactly two inputs.
3. (9 points) $N, M \leq 1000, Q \leq 5$
4. (4 points) $M = N + 1, M = 2^z$ (for some positive integer z), $P[i] = \lfloor \frac{i-1}{2} \rfloor$ (for each i such that $1 \leq i \leq N + M - 1$), $L = R$
5. (12 points) $M = N + 1, M = 2^z$ (for some positive integer z), $P[i] = \lfloor \frac{i-1}{2} \rfloor$ (for each i such that $1 \leq i \leq N + M - 1$)
6. (27 points) Each threshold gate has exactly two inputs.
7. (28 points) $N, M \leq 5000$
8. (11 points) No additional constraints.

Sample Grader

The sample grader reads the input in the following format:

- line 1: $N M Q$
- line 2: $P[0] P[1] \dots P[N + M - 1]$
- line 3: $A[0] A[1] \dots A[M - 1]$
- line $4 + k$ ($0 \leq k \leq Q - 1$): $L R$ for update k

The sample grader prints your answers in the following format:

- line $1 + k$ ($0 \leq k \leq Q - 1$): the return value of `count_ways` for update k