วงจรใหญ่ ใจกระพริบ

วงจรดิจิทัลประกอบด้วย **เกต (gate)** จำนวน $N + M$ เกต มีหมายเลขตั้งแต่ 0 ถึง $N + M - 1$ ให้เกต 0 ถึง $N - 1$ เป็น **เกตตัดสิน (threshold gates)** และเกต N ถึง $N + M - 1$ เป็น **เกตต้นกำเนิด (source gates)**

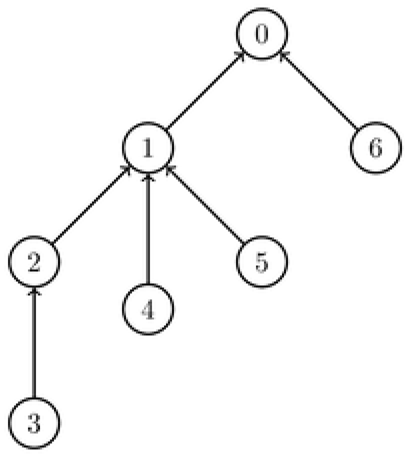
ทุกเกตยกเว้นเกต 0 ล้วนเป็น **อินพุต** ให้กับเกตตัดสินเพียงหนึ่งเกต กล่าวคือ สำหรับทุก i ที่ $1 \leq i \leq N + M - 1$ เกต i เป็นอินพุตให้กับเกต $P[i]$ เมื่อ $0 \leq P[i] \leq N - 1$ สิ่งสำคัญคือเรากำหนดให้ $P[i] < i$ นอกจากนี้ สมมติว่า $P[0] = -1$ เกตตัดสินแต่ละเกตมีอินพุตหนึ่งตัวขึ้นไป ส่วนเกตต้นกำเนิดไม่มีอินพุต

แต่ละเกตมี **สถานะ** เป็น 0 หรือ 1 อย่างใดอย่างหนึ่ง สถานะเริ่มต้นของเกตต้นกำเนิดระบุโดยอาร์เรย์ A ที่มีจำนวนเต็ม M ตัว กล่าวคือ สำหรับทุก j ที่ $0 \leq j \leq M - 1$ สถานะเริ่มต้นของเกตตัดสิน $N + j$ คือ $A[j]$

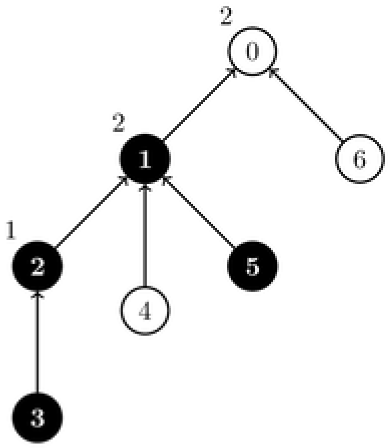
สถานะของเกตตัดสินแต่ละเกตขึ้นอยู่กับสถานะของอินพุตของเกตนั้น และจะมีการคำนวณสถานะดังนี้ เกตตัดสินแต่ละเกตจะถูกกำหนดค่า **พารามิเตอร์**ตัดสิน พารามิเตอร์ของเกตตัดสินที่มีอินพุตจำนวน c ตัว จะต้องเป็นจำนวนเต็มตั้งแต่ 1 ถึง c ดังนั้น สถานะของเกตตัดสินที่มีพารามิเตอร์ p จะมีค่าเป็น 1 ถ้าอินพุตอย่างน้อย p ตัวมีค่าเป็น 1 สถานะของเกตตัดสินจะมีค่าเป็น 0 ในกรณีอื่น

ยกตัวอย่างเช่น มีเกตตัดสิน $N = 3$ เกต และเกตต้นกำเนิด $M = 4$ เกต อินพุตของเกต 0 คือเกต 1 และเกต 6 อินพุตของเกต 1 คือเกต 2, เกต 4 และเกต 5 ส่วนอินพุตของเกต 2 มีตัวเดียวคือเกต 3

ตัวอย่างนี้แสดงได้ดังรูป



สมมติว่าเกตต้นกำเนิด 3 และ 5 มีสถานะเป็น 1 ขณะที่เกตต้นกำเนิด 4 และ 6 มีสถานะเป็น 0 หากเรากำหนดพารามิเตอร์ 1, 2 และ 2 ให้กับเกตตัดสิน 2, 1 และ 0 ตามลำดับ จะพบว่าเกต 2 มีสถานะเป็น 1, เกต 1 มีสถานะเป็น 1 และเกต 0 มีสถานะเป็น 0 การกำหนดค่าพารามิเตอร์และสถานะดังกล่าวได้แสดงไว้ดังรูปต่อไปนี้ ทั้งนี้เกตที่มีสถานะเป็น 1 จะถูกระบายด้วยสีดำ



สถานะของเกตต้นกำเนิดจะถูกอัปเดต Q รอบ การอัปเดตแต่ละครั้งกำหนดด้วยจำนวนเต็มสองตัวคือ L และ R ($N \le L \le R \le N + M - 1$) โดยจะเป็นการสลับสถานะของเกตต้นกำเนิดทุกตัวที่มีหมายเลขตั้งแต่ L ถึง R (รวมหัวท้าย) กล่าวคือ สำหรับทุก i ที่ $L \le i \le R$ เกตต้นกำเนิด i จะถูกเปลี่ยนสถานะให้เป็น 1 ถ้าสถานะเดิมเป็น 0, หรือถูกเปลี่ยนสถานะให้เป็น 0 ถ้าสถานะเดิมเป็น 1
เกตที่ถูกสลับจะยังคงสถานะใหม่ไว้โดยไม่เปลี่ยนแปลงจนกระทั่งจะถูกสลับอีกครั้งในการอัปเดตครั้งต่อ ๆ ไป

เป้าหมายของคุณคือการนับจำนวนรูปแบบการกำหนดพารามิเตอร์ที่แตกต่างกันของเกตตัดสินที่จะทำให้เกต 0 มีสถานะเป็น 1 หลังจากการอัปเดตในแต่ละครั้ง รูปแบบการกำหนดพารามิเตอร์สองรูปแบบจะแตกต่างกัน เมื่อมีเกตตัดสินอย่างน้อยหนึ่งเกตที่กำหนดค่าพารามิเตอร์ไว้แตกต่างกัน เนื่องจากอาจจะมีรูปแบบที่แตกต่างกันเป็นจำนวนมาก คุณควรจะคำนวณผลลัพธ์เป็นเศษของการหารด้วย $1\;000\;002\;022$

สังเกตว่าในตัวอย่างที่ให้มา จะมีรูปแบบการกำหนดพารามิเตอร์ให้เกตตัดสินที่แตกต่างกันทั้งหมด 6 รูปแบบเนื่องจากเกต 0, 1 และ 2 มีจำนวนอินพุต 2, 3 และ 1 ตัวตามลำดับ โดยใน 2 จาก 6 รูปแบบนี้ เกต 0 จะมีสถานะเป็น 1

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนสองฟังก์ชันต่อไปนี้

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

- N: จำนวนเกตตัดสิน
- M: จำนวนเกตต้นกำเนิด
- P: อาร์เรย์ขนาด $N + M$ ที่กำหนดค่าอินพุตสำหรับเกตตัดสิน
- A: อาร์เรย์ขนาด M ที่กำหนดสถานะเริ่มต้นของเกตต้นกำเนิด
- ฟังก์ชันนี้จะถูกเรียกเพียงครั้งเดียว ก่อนการเรียก `count_ways`

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

- L, R: กำหนดขอบเขตสำหรับช่วงของเกตต้นกำเนิดที่จะถูกสลับสถานะ
- ฟังก์ชันนี้จะต้องอัปเดตสถานะตามที่กำหนด จากนั้นจึงคืนค่าเป็นจำนวนรูปแบบการกำหนดพารามิเตอร์ของเกตตัดสินที่จะทำให้เกต 0 มีสถานะเป็น 1 โดยคืนค่าเป็นเศษของการหารด้วย $1\;000\;002\;022$

- ฟังก์ชันนี้จะถูกเรียกใช้ Q ครั้ง

ตัวอย่าง

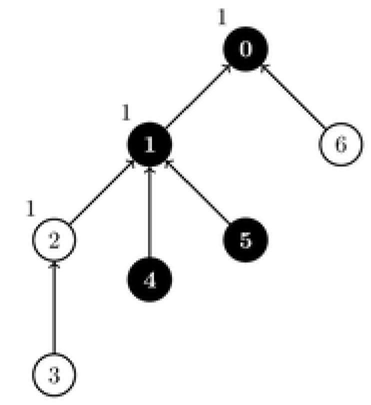
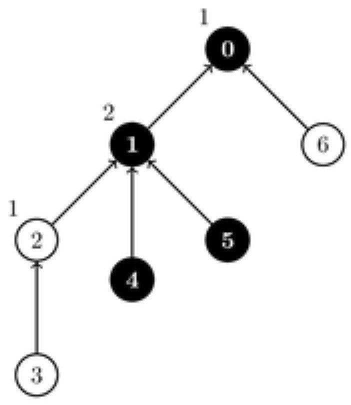
พิจารณาลำดับการเรียกฟังก์ชันต่อไปนี้:

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

ตัวอย่างนี้ได้ถูกอธิบายไว้ในรายละเอียดของโจทย์ข้างต้น

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

การเรียกฟังก์ชันนี้จะอัปเดตสถานะของเกต 3 และเกต 4 กล่าวคือ สถานะของเกต 3 จะกลายเป็น 0 และสถานะของเกต 4 จะกลายเป็น 1 โดยรูปแบบการกำหนดพารามิเตอร์สองรูปแบบที่ทำให้เกต 0 มีสถานะเป็น 1 จะแสดงอยู่ในภาพต่อไปนี้

รูปแบบที่ 1	รูปแบบที่ 2
	

ส่วนถ้าเป็นรูปแบบอื่น เกต 0 จะมีสถานะเป็น 0 ดังนั้นฟังก์ชันนี้ควรคืนค่าเป็น 2

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

การเรียกฟังก์ชันนี้จะอัปเดตสถานะของเกต 4 และเกต 5 ผลดังกล่าวจะทำให้เกตต้นกำเนิดทั้งหมดมีสถานะเป็น 0 ซึ่งไม่ว่าจะกำหนดรูปแบบพารามิเตอร์อย่างไรก็ตาม เกต 0 จะมีสถานะเป็น 0 ดังนั้นฟังก์ชันนี้ควรคืนค่าเป็น 0

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

การเรียกฟังก์ชันนี้จะเปลี่ยนสถานะของเกตต้นกำเนิดทั้งหมดเป็น 1 ผลดังกล่าวจะทำให้เกต 0 มีสถานะเป็น 1 ไม่ว่าจะกำหนดรูปแบบพารามิเตอร์อย่างไรก็ตาม ดังนั้น ฟังก์ชันนี้ควรคืนค่าเป็น 6

ข้อจำกัด

- $1 \leq N, M \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $P[0] = -1$
- $0 \leq P[i] < i$ และ $P[i] \leq N-1$ (สำหรับทุก i ที่ $1 \leq i \leq N+M-1$)
- เกตตัดสินแต่ละเกตจะมีอินพุตอย่างน้อยหนึ่งตัว (สำหรับทุก i ที่ $0 \leq i \leq N-1$ จะมีค่าดัชนี x ที่ $i < x \leq N+M-1$ และ $P[x] = i$)
- $0 \leq A[j] \leq 1$ (สำหรับทุก j ที่ $0 \leq j \leq M-1$)
- $N \leq L \leq R \leq N+M-1$

ปัญหาย่อย

1. (2 คะแนน) $N = 1$, $M \leq 1000$, $Q \leq 5$
2. (7 คะแนน) $N, M \leq 1000$, $Q \leq 5$, เกตตัดสินแต่ละตัวมีอินพุตสองตัวเท่านั้น
3. (9 คะแนน) $N, M \leq 1000$, $Q \leq 5$
4. (4 คะแนน) $M = N+1$, $M = 2^z$ (สำหรับจำนวนเต็มบวก z), $P[i] = \lfloor \frac{i-1}{2} \rfloor$ (สำหรับทุก i ที่ $1 \leq i \leq N+M-1$), $L = R$
5. (12 คะแนน) $M = N+1$, $M = 2^z$ (สำหรับจำนวนเต็มบวก z), $P[i] = \lfloor \frac{i-1}{2} \rfloor$ (สำหรับทุก i ที่ $1 \leq i \leq N+M-1$)
6. (27 คะแนน) เกตตัดสินทุกตัวมีอินพุตจำนวน 2 ตัว
7. (28 คะแนน) $N, M \leq 5000$
8. (11 คะแนน) ไม่มีข้อจำกัดเพิ่มเติม

เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างจะอ่านอินพุตตามรูปแบบต่อไปนี้:

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

เกรดเดอร์ตัวอย่างจะพิมพ์ผลลัพธ์ของคุณตามรูปแบบต่อไปนี้:

- บรรทัดที่ $1+k$ ($0 \leq k \leq Q-1$): ค่าที่ได้จากการเรียก `count_ways` สำหรับการอัปเดตรอบที่ k