

ความปลอดภัยของอาณาจักร

มีอาณาจักรแห่งหนึ่ง ประกอบด้วยเมือง N เมือง แต่ละเมืองเชื่อมกันอยู่ด้วยถนน M เส้น ถนนเส้นที่ j จะเชื่อมระหว่างเมือง u_j กับ v_j โดยรับประกันว่าภายในอาณาจักรนี้ทุกคูเมืองจะสามารถมีทางให้เดินไปหากันได้ แต่ละเมืองจะมีหมายเลขกำกับ s_i อยู่แทนค่าความแข็งแกร่งของเมืองที่ i

โชคร้าย ที่ล่าสุดมีกลุ่มผู้ก่อความไม่สงบ ต้องการก่อความวุ่นวายภายในอาณาจักรนี้ โดยกลุ่มผู้ก่อความไม่สงบจะดำเนินการสร้างความวุ่นวายให้กับอาณาจักรดังนี้

ผู้ก่อความไม่สงบจะทำการเลือกค่า a และ b มา และจะทำการโจมตีไปเรื่อยๆ หลายวัน เริ่มจากวันที่ 1 จะมีค่าพลัง $a + b$ หน่วย ในแต่ละวันค่าพลังจะเพิ่มขึ้นเรื่อย ๆ a หน่วย กล่าวคือ ในแต่ละวันผู้ก่อความไม่สงบจะมีค่าพลัง $a + b, 2a + b, 3a + b, \dots$ ตามลำดับ

ในแต่ละวัน ผู้ก่อความไม่สงบสามารถเลือกเมืองหนึ่งเมือง (สมมติเป็นเมือง x) ที่สอดคล้องกับเงื่อนไขดังต่อไปนี้:

- เมือง x นั้นยังไม่เคยโดนโจมตีมาก่อน
- ค่าความแข็งแกร่ง s_x นั้นมีค่าน้อยกว่าหรือเท่ากับค่าพลังของผู้ก่อความไม่สงบในวันนั้น
- มีถนนที่เชื่อมระหว่างเมือง x กับเมืองอื่นที่ยังไม่โดนโจมตีเพียงแค่เส้นเดียวหรือศูนย์เส้นเท่านั้น

เมื่อเลือกเมือง x แล้ว ผู้ก่อความไม่สงบจะโจมตีเมือง x ไปได้ และจะถือว่าหมดวัน (การโจมตีแต่ละครั้งใช้เวลา 1 วันพอดี) โดยผู้ก่อความไม่สงบสามารถวางแผนโจมตีนี้ได้อย่างไรก็ได้ จนกระทั่งวันไหนก็ตามที่ไม่มีเมืองใดที่สอดคล้องกับเงื่อนไขเหล่านี้เลย จะถือว่าผู้ก่อความไม่สงบนั้นพ่ายแพ้ต่อระบบการป้องกันของอาณาจักรในวันนั้น ยกเว้นว่าผู้ก่อความไม่สงบนั้นได้โจมตีไปครบทุกเมืองแล้ว จะถือว่าชนะ

นักวิเคราะห์ความปลอดภัยของอาณาจักรต้องการถามคำถามทั้งหมด Q คำถาม ในคำถามที่ k จะให้ค่า a_k มา แล้วถามว่าผู้ก่อความไม่สงบจะต้องเลือกค่า $b_k \geq 0$ อย่างน้อยที่สุดเท่าไรจึงจะสามารถสร้างชนะการป้องกันของอาณาจักร (หากไม่รู้ว่าเลือก b_k มากเท่าไรก็ทำไม่ได้ ให้ตอบ -1 หากเลือก $b_k = 0$ ก็ทำได้ ให้ตอบ 0 และไม่ต้องคำนึงถึงกรณีที่ $b_k < 0$) โดยจะเป็นการโจมตีตามรูปแบบที่กล่าวไปข้างต้น เมื่อแทน $a = a_k$ และ $b = b_k$ (นอกจากนี้รับประกันว่า $a_k \geq 0$)

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันดังต่อไปนี้

```
void kingdom(int N, int M, int Q, vector<int> u, vector<int> v, vector<int> s)
```

- ฟังก์ชันนี้จะถูกเรียกเพียงครั้งเดียว

- รับประกันว่า $0 \leq u_j, v_j < N$ และ $u_j \neq v_j$ สำหรับทุก j ระหว่าง 0 ถึง $M - 1$
- รับประกันว่า $1 \leq s_i \leq 1\,000\,000\,000$ สำหรับทุก i ระหว่าง 0 ถึง $N - 1$

```
long long answer_query(int a)
```

- ฟังก์ชันนี้จะถูกเรียกทั้งหมด Q ครั้ง
- ในครั้งที่ k (สำหรับ $0 \leq k < Q$) ค่า $a[k]$ คือ a_k ของคำถามนั้น
- จะต้องคืนค่า $b_k \geq 0$ ที่น้อยที่สุดที่เป็นไปได้ หากเป็นไปได้ให้ตอบ -1
- รับประกันว่า $0 \leq a_k \leq 1\,000\,000\,000$

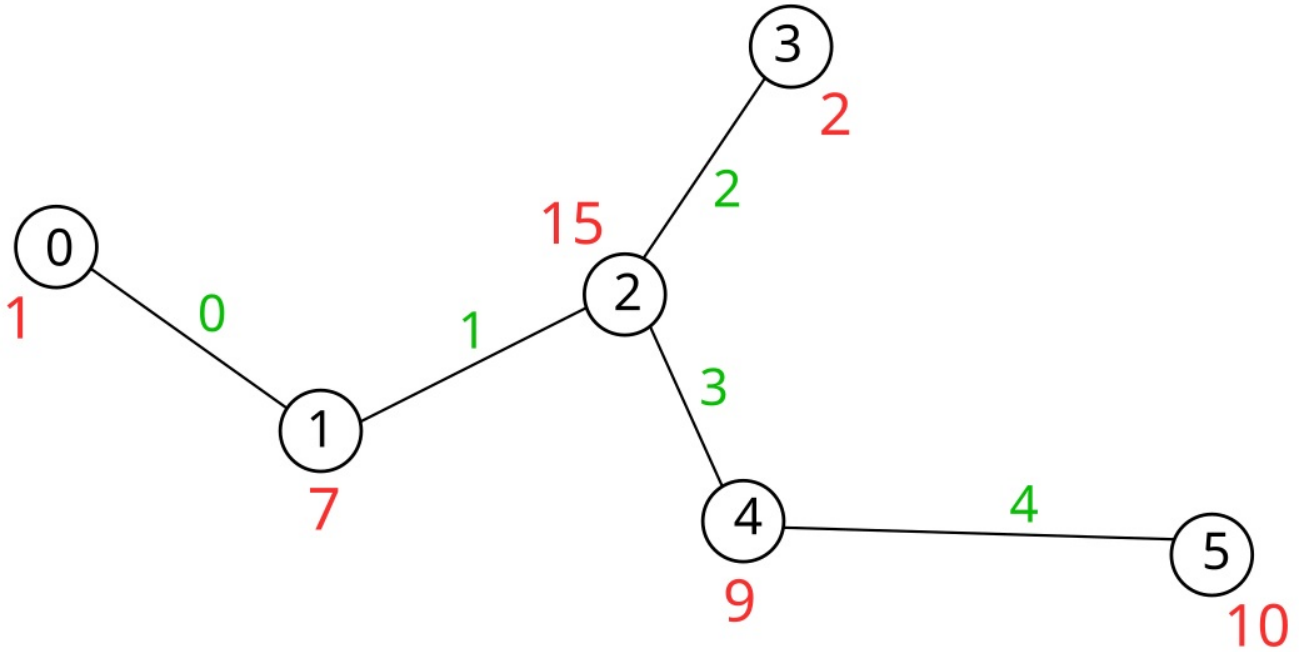
ข้อจำกัด

- $2 \leq N, Q \leq 800\,000$
- $1 \leq M \leq 1\,000\,000$

ปัญหาย่อย

1. (5 คะแนน) $M \geq N$
2. (15 คะแนน) $N, M, Q \leq 2000$
3. (10 คะแนน) s_i นั้นมีค่าเท่ากันทั้งหมด สำหรับทุก i ระหว่าง 0 ถึง $N - 1$
4. (15 คะแนน) $M = N - 1; u_j = j; v_j = j + 1$ สำหรับทุก j ระหว่าง 0 ถึง $M - 1$ นอกจากนี้ $(s_i)_{i=1}^N$ จะเรียงจากน้อยไปมาก
5. (10 คะแนน) $M = N - 1; u_j = j; v_j = j + 1$ สำหรับทุก j ระหว่าง 0 ถึง $M - 1$
6. (25 คะแนน) $N, Q \leq 90\,000$ และ $M \leq 100\,000$
7. (20 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง



```
kingdom(6, 5, 2, [0, 1, 2, 2, 4], [1, 2, 3, 4, 5], [1, 7, 15, 2, 9, 10])
```

ถัดมา จะมีการเรียก `answer_query` สองครั้ง ดังนี้

```
answer_query(2)
```

ฟังก์ชันจะต้องคืนค่า 3 ซึ่งเป็นคำตอบที่ถูกต้อง

ตัวอย่างของการโจมตีที่ผู้ก่อความไม่สงบสามารถโจมตีด้วย $a = 2$ และ $b = 3$ ได้เป็นดังต่อไปนี้

- ในวันที่ 1 มีค่าพลัง $2 + 3 = 5$ จึงสามารถทำการโจมตีเมือง 3 ซึ่งมีความแข็งแกร่ง 2 ได้
- ต่อมาวันที่ 2 มีค่าพลัง $2 + 5 = 7$ จึงสามารถทำการโจมตีเมือง 0 ซึ่งมีความแข็งแกร่ง 1 ได้
- ต่อมาวันที่ 3 มีค่าพลัง $2 + 7 = 9$ จึงสามารถทำการโจมตีเมือง 1 ซึ่งมีความแข็งแกร่ง 7 ได้
- ต่อมาวันที่ 4 มีค่าพลัง $2 + 9 = 11$ จึงสามารถทำการโจมตีเมือง 5 ซึ่งมีความแข็งแกร่ง 10 ได้
- ต่อมาวันที่ 5 มีค่าพลัง $2 + 11 = 13$ จึงสามารถทำการโจมตีเมือง 4 ซึ่งมีความแข็งแกร่ง 9 ได้
- สุดท้ายวันที่ 6 มีค่าพลัง $2 + 13 = 15$ จึงสามารถทำการโจมตีเมือง 2 ซึ่งมีความแข็งแกร่ง 15 ได้

สังเกตว่า หาก $b < 3$ จะไม่มีวิธีการใดที่เป็นไปได้

```
answer_query(1)
```

ฟังก์ชันจะต้องคืนค่า 9 ซึ่งเป็นคำตอบที่ถูกต้อง

เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างจะอ่านข้อมูลดังต่อไปนี้:

- บรรทัดที่ 1: $N \ M \ Q$
- บรรทัดที่ 2: $s[0] \ s[1] \ s[2] \ \dots \ s[N - 1]$
- บรรทัดที่ $3 + j$ (ถึง $3 + M - 1$): $u[j] \ v[j]$
- บรรทัดที่ $3 + M + k$ (ถึง $3 + M + Q - 1$): $a[k]$

ขอบเขต

- Time limit: 4 seconds
- Memory limit: 512 MB