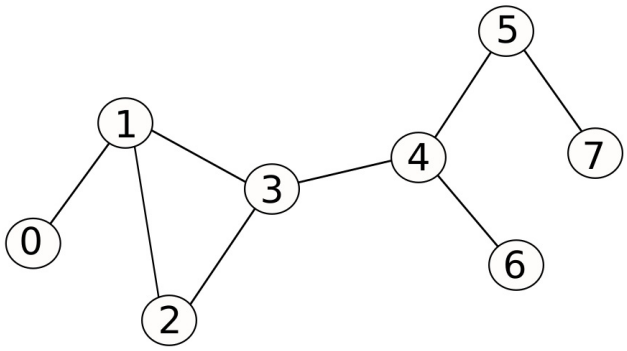
เส้นทางเดียว (onepath)

ประเทศแห่งหนึ่งมี N จังหวัด เรียกเป็นจังหวัดที่ 0 ถึง $N-1$ เมื่อเริ่มต้นมีถนนแบบที่เดินทางได้สองทิศทางเชื่อมระหว่างจังหวัดเหล่านี้ จำนวน M เส้น ถนนแต่ละเส้นจะเชื่อมระหว่างสองจังหวัด ระหว่างคู่ของจังหวัดใด ๆ จะมีถนนเชื่อมไม่เกิน 1 เส้น การเดินทางระหว่างจังหวัดจะใช้ถนนเหล่านี้ รับประกันว่าระหว่างคู่ของจังหวัดใด ๆ จะสามารถเดินทางหากันได้เสมอ

ประเทศแห่งนี้มีการปล้นธนาคารบ่อยมาก ทำให้การขนเงินระหว่างธนาคารข้ามจังหวัดจะต้องใช้ความระมัดระวังที่สูงกว่าปกติ โดยเฉพาะในกรณีที่มีเพียงเส้นทางเพียงเส้นทางเดียวที่เป็นไปได้ เราจะกล่าวว่าการเดินทางจากจังหวัด u ไปจังหวัด v นั้น *อันตราย* ถ้ามีเส้นทางที่ผ่านจังหวัดใด ๆ ไม่เกินหนึ่งครั้งจากจังหวัด u ไปยังจังหวัด v แค่หนึ่งเส้น คุณจะได้รับคำถามจำนวนหลายคำถามว่าการเดินทางระหว่างจังหวัดนั้นอันตรายหรือไม่

นอกจากนี้ แม้ว่าผู้คนจะใช้ชีวิตอยู่ใต้ความเสี่ยง งบประมาณในการปรับปรุงเชื่อมต่อถนนก็ยังมีอยู่ตลอดเวลา ดังนั้นนอกจากที่คุณจะได้รับคำถามแล้ว คุณยังจะได้รับข้อมูลว่ามีการเพิ่มถนนเชื่อมระหว่างคู่ของจังหวัดอีกด้วย คุณจะได้รับคำถามและคำสั่งระบุว่ามีการเพิ่มถนนรวมกันทั้งสิ้น Q ครั้ง

พิจารณากรณีตัวอย่างที่ $N = 8$ และ $M = 8$ ต่อไปนี้



ที่จุดเริ่มต้นนี้ การเดินทางจากจังหวัด 7 ไปยัง 6 นั้นอันตราย หรือการเดินทางจากจังหวัด 3 ไปยังจังหวัด 5 ก็อันตราย อย่างไรก็ตามการเดินทางจากจังหวัด 0 ไป 6 นั้น **ไม่**อันตราย เพราะว่ามีเส้นทางมากกว่าหนึ่งเส้นทางที่ผ่านจังหวัดใด ๆ ไม่เกินหนึ่งครั้ง คือ

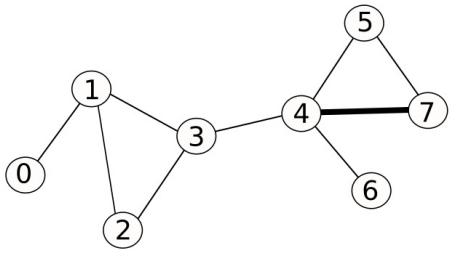
$$0 - 1 - 3 - 4 - 6$$

และ

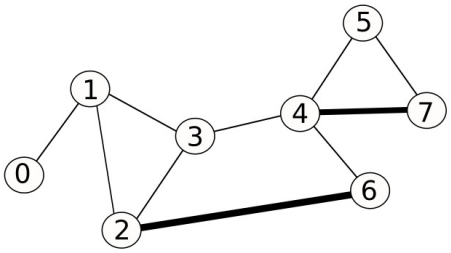
$$0 - 1 - 2 - 3 - 4 - 6$$

ในทำนองเดียวกันการเดินทางจากจังหวัด 2 ไปยัง 7 ก็ไม่อันตรายเช่นเดียวกัน

สมมติว่ามีการเพิ่มถนนระหว่างจังหวัด 4 กับ 7 ดังแสดงในรูปล่างซ้าย (รูป (a))



(a)



(b)

การเดินทางจากจังหวัด 3 ไปยัง 5 จะไม่อันตรายแล้วเพราะว่ามีเส้นทางสองเส้นทาง อย่างไรก็ตามการเดินทางจากจังหวัด 3 ไปยัง 6 จะยังอันตรายอยู่

สุดท้ายถ้ามีการเพิ่มถนนระหว่างจังหวัด 2 กับ 6 ดังแสดงในรูปขวาด้านบน (รูป (b)) การเดินทางจากจังหวัด 3 ไปยัง 6 จะไม่อันตรายแล้ว

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชัน 3 ฟังก์ชันต่อไปนี้

```
void initialize(int N, int M, int Q, vector<pair<int,int>> R)
```

- ฟังก์ชันนี้จะถูกเรียกหนึ่งครั้ง โดยที่ N แทนจำนวนจังหวัด M แทนจำนวนถนนเมื่อเริ่มต้นและ Q แทนจำนวนคำถามและคำสั่ง
- อาร์เรย์ R ที่มีขนาด M จะระบุข้อมูลถนนเริ่มต้น กล่าวคือ สำหรับ $0 \leq i < M$ ถนนที่ i จะเชื่อมระหว่างจังหวัด $R[i][0]$ กับ $R[i][1]$ รับประกันว่าตลอดการทำงานจะไม่มีคู่ของจังหวัดใดมีถนนเชื่อมมากกว่าหนึ่งเส้น

หลังจากนั้นเกรดเดอร์จะเรียกฟังก์ชันด้านล่าง รวม Q ครั้ง โดยจะเป็นคำถาม และคำสั่งในการสร้างถนน

```
bool is_dangerous(int u, int v)
```

- ฟังก์ชันนี้จะต้องตอบคำถามว่าการเดินทางจากจังหวัด u ไปจังหวัด v นั้นอันตรายหรือไม่ (กล่าวคือ มีเส้นทางที่ไม่ผ่านจังหวัดใด ๆ มากกว่าหนึ่งครั้งเพียงเส้นทางเดียวระหว่างจังหวัดทั้งสองหรือไม่) โดยคืนค่าเป็น `true` หรือ `false`

```
void build_road(int u, int v)
```

- ฟังก์ชันนี้ระบุว่ามีการสร้างถนนเชื่อมจังหวัด u กับจังหวัด v เข้าด้วยกัน รับประกันว่าตลอดการทำงานระหว่างคู่ของจังหวัดใด ๆ จะมีถนนไม่เกินหนึ่งเส้น

Constraints

- $2 \leq N \leq 100\,000$
- $1 \leq M, Q \leq 200\,000$

- $0 \leq R[i][k] < N$ สำหรับทุก ๆ $0 \leq i < M$ และ $k \in \{0, 1\}$
- $0 \leq u < N, 0 \leq v < N, u \neq v$

Subtasks

1. (6 points) กราฟเส้นตรง ($R[i][0] = i, R[i][1] = i + 1$) แต่ละโหนดอยู่ไม่เกิน cycle เดียว
2. (7 points) กราฟเส้นตรง ($R[i][0] = i, R[i][1] = i + 1$)
3. (11 points) ต้นไม้ แต่ละโหนดอยู่ไม่เกิน cycle เดียว
4. (5 points) ต้นไม้ star ($R[i][0] = 0, R[i][1] = i + 1$)
5. (9 points) ฟังก์ชัน `build_road` จะถูกเรียกก่อน `is_dangerous` เสมอ, $N \leq 1000$
6. (12 points) ฟังก์ชัน `build_road` จะถูกเรียกก่อน `is_dangerous` เสมอ
7. (4 points) $N \leq 100$
8. (9 points) $N \leq 1000$
9. (37 points) ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

ด้านล่างแสดงการเรียกใช้ฟังก์ชันต่าง ๆ ตามตัวอย่างในโจทย์ด้านบนที่ $N = 8, M = 8, Q = 9$

```
initialize(8,8,9,
        [[0,1], [1,2], [3,1], [2,3], [4,3], [6,4], [4,5], [5,7]] )
```

จากนั้นเกรดเดอร์จะเรียกฟังก์ชัน `is_dangerous` เพื่อถามและคำตอบที่ถูกต้องควรเป็นดังนี้

คำถาม	คำตอบ
`is_dangerous(7,6)`	`true`
`is_dangerous(3,5)`	`true`
`is_dangerous(0,6)`	`false`
`is_dangerous(2,7)`	`false`

เกรดเดอร์จะเรียกฟังก์ชัน `build_road` เพื่อสร้างถนน

```
build_road(4,7)
```

จากนั้นเกรดเดอร์จะเรียกฟังก์ชัน `is_dangerous` เพื่อถามและคำตอบที่ถูกต้องควรเป็นดังนี้

คำถาม	คำตอบ
`is_dangerous(3,5)`	`false`
`is_dangerous(3,6)`	`true`

เกรดเดอร์จะเรียกฟังก์ชัน `build_road` เพื่อสร้างถนน

```
build_road(2,6)
```

จากนั้นเกรดเดอร์จะเรียกฟังก์ชัน `is_dangerous` เพื่อถามและคำตอบที่ถูกต้องควรเป็นดังนี้

คำถาม	คำตอบ
`is_dangerous(3,6)`	`false`

Sample Grader

The sample grader reads data as the following:

- Line 1: N M Q
- Line 2 ... $M+1$: $R[i][0]$ $R[i][1]$
- Line $M+2$ ... $M+Q+1$: t_i u_i v_i

โดยที่ $t_i = 1$ คือเรียก `is_dangerous` และ $t_i = 2$ คือเรียก `build_road`

Limits

- Time limit: 1 second
- Memory limit: 512 MB