

เมืองโดดเดี่ยว (lonely-town)

ในประเทศแห่งหนึ่งมีเมืองอยู่ N เมือง เรียกเป็นเมือง 0 ถึงเมือง $N - 1$ เพื่อที่จะเดินทางไปมาหาสู่กันได้ ประชาชนจึงได้ลงขันกันสร้างถนนปูนจำนวน $N - 1$ เส้น เพื่อเชื่อมต่อให้เมืองทุกเมืองเดินทางไปมาหาสู่กันได้ โดยถนนปูนเส้นที่ i (สำหรับ $0 \leq i < N - 1$) เชื่อมระหว่างเมือง $u[i]$ กับ $v[i]$ (รับประกันว่า $u[i] \neq v[i]$)

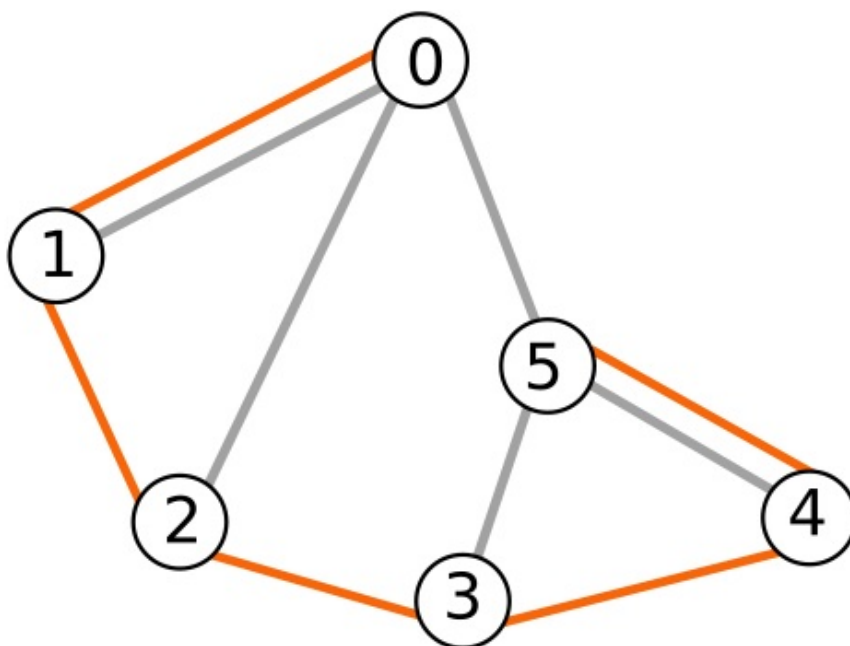
ทางการเห็นว่าการมีถนนเป็นสิ่งจำเป็น จึงได้ออกงบประมาณสร้างถนนหลวงที่แข็งแรงกว่าเพิ่มเติมอีก $N - 1$ เส้น ถนนหลวงเส้นที่ e สำหรับ $0 \leq e \leq N - 2$ จะเชื่อมระหว่างเมือง e และ $e + 1$ ระหว่างเมืองสองเมืองใด ๆ อาจจะมีถนนปูนหรือถนนหลวงเชื่อมมากกว่าหนึ่งเส้นก็ได้ แต่ตลอดการทำงานจะไม่มีถนนปูนเชื่อมระหว่างคู่ของเมืองใด ๆ มากกว่า 20 เส้น

เราจะกล่าวว่าเมือง i ใด ๆ เป็นเมือง**โดดเดี่ยว** ถ้าเมือง i ไม่สามารถเดินทางไปเมืองอื่น ๆ ได้เลย **หรือ** ไม่ว่าจะเดินทางไปเมืองใดก็ตาม ถ้าจะเดินทางวนกลับมาที่เมือง i จะต้องมีการใช้ถนนบางเส้นซ้ำกันเสมอ

ระหว่างการทำงาน จะมีคำสั่งจากรัฐบาลได้สามแบบ คือ

1. เปลี่ยนถนนปูนหนึ่งเส้นที่ติดกับเมือง x และเชื่อมกับเมือง y ให้ไปเชื่อมกับเมือง z แทน
2. สอบถามว่าปัจจุบันมีจำนวนเมืองโดดเดี่ยวกี่เมือง
3. แสดงความสำคัญของถนนหลวง โดย **จำลอง** การทุบถนนหลวงเส้นที่ e ทั้งและสอบถามจำนวนเมืองโดดเดี่ยว เมื่อเสร็จสิ้นกระบวนการ ถนนหลวงทั้งหมดจะกลับมาใช้งานได้เหมือนเดิม เพราะว่าเป็นแค่การจำลองสถานการณ์เท่านั้น

พิจารณาตัวอย่างที่ $N = 6$ ดังแสดงในรูปต่อไปนี้

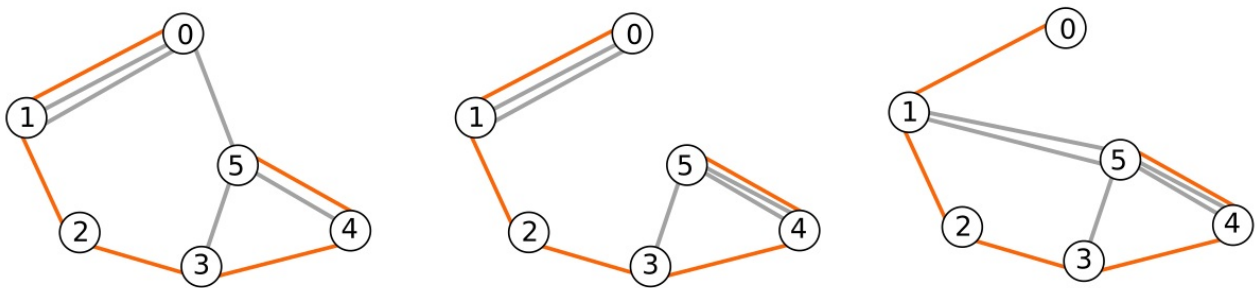


ถนนปูนเมื่อตอนเริ่มต้นแสดงเป็นถนนสีเทา ถนนหลวงแสดงเป็นถนนสีส้ม สังเกตว่าในตอนแรกนี้ ไม่มีเมืองใดเป็นเมืองโดดเดี่ยวเลย ดังนั้นถ้ามีคำสั่งแบบที่ 2 จากรัฐบาล คำตอบที่ได้คือ 0

ถ้ารัฐบาลส่งคำสั่งแบบที่ 1 ให้เปลี่ยนถนนปูนที่เชื่อมจากเมือง 0 ไป 2 ให้ไปเชื่อมกับเมือง 1 แทน ผลลัพธ์จะเป็นดังรูปด้านล่างซ้าย หลังคำสั่งนี้ ก็ยังไม่มีเมืองใดเป็นเมืองโดดเดี่ยวเลย ดังนั้นถ้ามีคำสั่งแบบที่ 2 จากรัฐบาล คำตอบที่ได้คือ 0

ถ้ารัฐบาลส่งคำสั่งแบบที่ 1 อีกคำสั่งให้เปลี่ยนถนนปูนที่เชื่อมจากเมือง 5 ไปยังเมือง 0 ให้ไปเชื่อมกับเมือง 4 ผลลัพธ์จะเป็นดังรูปด้านล่างกลาง ภายหลังคำสั่งนี้ สังเกตว่าเมือง 2 จะเป็นเมืองโดดเดี่ยว ดังนั้นถ้ามีคำสั่งแบบที่ 2 จากรัฐบาล คำตอบที่ได้คือ 1

ต่อมาถ้ารัฐบาลส่งคำสั่งแบบที่ 1 อีกสองคำสั่ง ให้เปลี่ยนถนนปูนที่เชื่อมจากเมือง 1 ไปยังเมือง 0 ให้ไปเชื่อมเมือง 5 และให้เปลี่ยนถนนปูนที่เชื่อมจากเมือง 1 ไปยังเมือง 0 ให้ไปเชื่อมเมือง 5 (ซ้ำอีกครั้ง) ผลลัพธ์ที่ได้จะเป็นดังรูปด้านล่างขวา จะพบว่าเมือง 0 จะเป็นเมืองโดดเดี่ยว แต่เมือง 1 และเมือง 2 ไม่ใช่เมืองโดดเดี่ยว ดังนั้นถ้ามีคำสั่งแบบที่ 2 จากรัฐบาล คำตอบที่ได้คือ 1



รัฐบาลสามารถส่งคำสั่งแบบที่ 3 เพื่อจำลองการทุบถนนหลวงได้ สมมติว่ารัฐบาลส่งคำสั่งให้จำลองการทุบถนนหลวงเส้นที่ 1 ที่เชื่อมระหว่างเมือง 1 และ 2 เราจะพบว่าเมือง 2 ตอนนี้ กลายเป็นเมืองโดดเดี่ยว ในขณะที่เมือง 1 และเมือง 3 ยังไม่ใช่เมืองโดดเดี่ยว (เพราะว่าเมือง 1 สามารถเดินทางไปเมือง 5 และกลับมาเมือง 1 ได้ โดยไม่ต้องใช้ถนนซ้ำเดิม เพราะว่ามีถนนเชื่อมสองเส้น, และเมือง 3 สามารถเดินทางไปเมือง 5 แล้ววนกลับมาทางเมือง 4 ได้ โดยไม่ต้องใช้ถนนซ้ำเช่นเดียวกัน) ดังนั้นผลลัพธ์ของคำสั่งนี้คือ 2 เพราะว่าเมือง 0 และเมือง 2 กลายเป็นเมืองโดดเดี่ยว หลังจากจบคำสั่งนี้ ถนนหลวงเส้นที่ 2 จะกลับมาใช้งานได้แบบเดิม เนื่องจากคำสั่งแบบที่ 3 เป็นแค่การจำลองเท่านั้น

ถ้าสุดท้ายมีการส่งคำสั่งแบบที่ 2 มาอีกครั้ง คำตอบจะเป็น 1

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันดังต่อไปนี้:

```
void init_town(int N, int Q, vector<int> u, vector<int> v)
```

- ฟังก์ชันนี้จะถูกเรียกใช้โดยเกรตเตอร์เพียงครั้งเดียว
- N แทนจำนวนเมืองทั้งหมด
- Q แทนจำนวนคำสั่งจากรัฐบาล โดยแต่ละคำสั่งจะไปเรียกใช้งานหนึ่งในสามฟังก์ชันด้านล่าง
- $u[i], v[i]$ แทนว่ามีถนนปูนที่เชื่อมต่อระหว่างเมือง $u[i]$ กับเมือง $v[i]$ ($0 \leq i \leq N - 1$)
- ฟังก์ชันนี้ไม่มีการคืนค่า

```
void change_edge(int x, int y, int z)
```

- x, y, z หมายถึงทำการเปลี่ยนถนนปูนเส้นใดเส้นหนึ่งของเมือง x ที่เชื่อมกับเมือง y อยู่ให้ไปเชื่อมกับเมือง z แทน
- ฟังก์ชันนี้ไม่มีการคืนค่า
- รับประกันว่ามีถนนปูนอยู่อย่างน้อย 1 เส้นที่เชื่อมเมือง x กับเมือง y อยู่จึงจะมีการเรียกใช้ฟังก์ชันนี้

```
int lonelytown_query()
```

- ฟังก์ชันนี้จะต้องคืนค่าจำนวนเมืองโดดเดี่ยวปัจจุบันทั้งหมด

```
int simulate_query(int e)
```

- e หมายถึงจำลองทุบถนนหลวงเส้นที่ e ทิ้ง
- ฟังก์ชันนี้จะต้องคืนค่าจำนวนเมืองโดดเดี่ยวปัจจุบันทั้งหมด
- เมื่อฟังก์ชันนี้จบลง ถนนหลวงเส้นที่ e จะกลับมาเหมือนเดิมเนื่องจากการจำลอง

เงื่อนไข

- $2 \leq N, Q \leq 100\,000$
- $0 \leq u_i, v_i, x, y, z \leq N - 1$ และ $u_i \neq v_i$
- $0 \leq e \leq N - 2$
- เมืองสองเมืองใดๆจะมีถนนปูนเชื่อมต่อกันไม่เกิน 20 เส้น

ปัญหาย่อย

1. (9 คะแนน) $N \leq 2\,000, Q \leq 300$ และ ไม่มีการเรียกใช้ `simulate_query`
2. (15 คะแนน) $N \leq 500$
3. (12 คะแนน) $N \leq 2\,000$ และ ไม่มีการเรียกใช้ `simulate_query`
4. (16 คะแนน) `change_edge` จะถูกเรียกก่อน `lonelytown_query` และ `simulate_query` เสมอ
5. (21 คะแนน) ไม่มีการเรียกใช้ `simulate_query`
6. (27 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

จากตัวอย่างข้างต้น เกรตเดอร์จะเรียก

```
init_town(6, 10, [0,2,0,3,5], [1,0,5,5,4])
```

หลังจากนั้นจะมีการส่งคำสั่งทั้งสิ้น 10 ครั้งดังนี้ รัฐบาลจะส่งคำสั่งแบบที่ 2 เพื่อถามจำนวนเมืองโดดเดี่ยว คำตอบคือ 0

```
lonelytown_query() // ตอบ 0
```

จากนั้นจะส่งคำสั่งแบบที่ 1 เพื่อเปลี่ยนถนนปูนที่เชื่อมจากเมือง 0 ไปเมือง 2 ให้ไปเมือง 1 แทน และถ้ามีการส่งคำสั่งแบบที่ 2 คำตอบคือ 0

```
change_edge(0,2,1)
lonelytown_query() // ตอบ 0
```

จากนั้นจะส่งคำสั่งแบบที่ 1 เพื่อเปลี่ยนถนนปูนที่เชื่อมจากเมือง 5 ไปเมือง 0 ให้ไปเมือง 4 แทน และถ้ามีการส่งคำสั่งแบบที่ 2 คำตอบคือ 1

```
change_edge(5,0,4)
lonelytown_query() // ตอบ 1
```

หลังจากนั้น รัฐบาลอาจส่งคำสั่งแบบที่ 1 สองคำสั่งเปลี่ยนถนนปูนจากเมือง 1 ไป 0 ให้ไปเชื่อมเมือง 5 (ซ้ำกันสองครั้ง) และถ้ามีการส่งคำสั่งแบบที่ 2 คำตอบคือ 1

```
change_edge(1,0,5)
change_edge(1,0,5)
lonelytown_query() // ตอบ 1
```

รัฐบาลสามารถส่งคำสั่งแบบที่ 3 เพื่อจำลองการทุบถนนหลวงได้ โดยถ้าสั่งให้จำลองการทุบถนนหลวงเส้นที่ 1 ผลลัพธ์ที่ได้จะเท่ากับ 2

```
simulate_query(1) // ตอบ 2
```

ถนนหลวงจะไม่มี การเปลี่ยนแปลงหลังการจำลอง ดังนั้นถ้ามีการส่งคำสั่งแบบที่ 2 อีกครั้ง คำตอบที่ได้จะเท่ากับ 1 เหมือนเดิม

```
lonelytown_query() // ตอบ 1
```

เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างจะอ่านข้อมูลนำเข้าดังนี้

- บรรทัดที่ 1: $N \ Q$
- $N - 1$ บรรทัดถัดมา: $u[i] \ v[i]$
- Q บรรทัดถัดมา: ระบุคำสั่ง Q ครั้งในรูปแบบดังนี้:
 - $t(1) : x, y, z$
 - $t(2)$
 - $t(3) : e$

เกรตเตอร์ตัวอย่างจะส่งออกมาที่ได้รับจาก lonelytown_query และ simulate_query

ขีดจำกัด

- Time limit: 1.5 seconds
- Memory limit: 256 MB