



อำนาจมืด

"อำนาจมืด นี่ไง ปิดไฟ มืดแล้ว" — นักทฤษฎีผู้ไม่ได้พิสูจน์ทฤษฎีที่น่าสนใจมานานกว่าสิบปี, 2019

ในดินแดนราชอาณาจักรสารขันธ์ มีค่ายคอมมิวนิสต์ ซึ่งมีคนอยู่ 200 000 คน นำโดยแนวคิดของนักทฤษฎี วานอม-อีธาน ที่ยึดถือแนวปฏิบัติของการให้เกียรติกันและกัน และเป็นเสรีนิยมสุดโต่ง ทำให้ราชอาณาจักรสารขันธ์เป็นดินแดนสวรรค์นั่นเอง

อย่างไรก็ตาม ในค่ายคอมมิวนิสต์นี้ได้เกิดการโง่กันขึ้น ประเด็นสำคัญคือคนในค่ายพยายามปกปิดการโง่ โดยใช้สุภาษิตที่ว่า "โกเห็นดินสูง ฐเห็นนมโก" กล่าวคือต่างฝ่ายต่างพยายามหาข้อมูลการโง่ของคนอื่นไว้ เพื่อใช้เป็นข้อต่อรองให้ไม่มีใครเอาการโง่ไปแจ้งต่อ วานอม-อีธาน

เราจะเรียกการที่คนสองคนทำการโง่กันทั้งคู่ และต่างฝ่ายต่างรับรู้การโง่ของกันและกันนั้นว่าเหตุการณ์ "โกเห็นดินสูง ฐเห็นนมโก" (ขอเรียกสั้น ๆ ว่า เหตุการณ์โก่ง) เมื่อเกิดเหตุการณ์นี้ขึ้น คนสองคนนั้นก็เลยแลกเปลี่ยนข้อมูลการโง่ทั้งหมดเท่าที่ตัวเองรู้มา และหากสองคนนี้รับรู้ข้อมูลการโง่อื่นใดเพิ่มเติมหลังจากเหตุการณ์โก่ง ก็จะแลกเปลี่ยนข้อมูลกันไปตลอดด้วย

การแลกเปลี่ยนข้อมูลมีความหมายดังนี้

- หาก person1 รับรู้การโง่ของ person2 จะได้ว่า person2 ก็รับรู้ถึงการโง่ของ person1
- หาก person1 รับรู้การโง่ของ person2 และ person2 รับรู้การโง่ของ person3 จะได้ว่า person1 ก็รับรู้การโง่ของ person3

จงเขียนโปรแกรมรับข้อมูลว่า คนในค่ายคอมมิวนิสต์นี้ เกิดเหตุการณ์โก่งขึ้น ณ เวลาใดบ้าง และหาคำตอบว่า คู่ของคนสองคนในค่ายคอมมิวนิสต์นี้รับรู้การโง่ของกันและกันเร็วสุดเมื่อไร

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

```
void cheat(int person1, int person2)
```

- ฟังก์ชันนี้จะถูกเรียกใช้งาน N ครั้ง
- สำหรับการเรียกฟังก์ชันนี้ในครั้งที่ T จะหมายความว่าเกิด เหตุการณ์โก่ง ระหว่าง person1 และ person2 ในวินาทีที่ T ซึ่งทั้งคู่จะถูกนับว่าโง่โดยปริยายทันที (ถ้าทั้งคู่ยังไม่มีการโง่กันมาก่อนหน้านี้) และแลกเปลี่ยนข้อมูลกัน (รวมถึงแลกเปลี่ยนข้อมูลกันไปเรื่อย ๆ ในอนาคตด้วย) กล่าวคือ ไม่ว่า person2 จะรู้ว่าใครโง่ person1 ก็จะได้รู้ด้วย และเช่นเดียวกัน ไม่ว่า person1 จะรู้ว่าใครโง่ person2 จะรู้ด้วย และในอนาคต หากมีใครรับรู้การโง่เพิ่มขึ้น อีกคนก็จะรับรู้ไปด้วย
- เป็นไปได้ที่ฟังก์ชันนี้จะถูกเรียกซ้ำกันโดยมีค่า person1 และ person2 เป็นคู่คนที่เคยเกิด เหตุการณ์โก่งมาแล้ว ในกรณีนี้ให้นับว่าเวลาผ่านไปหนึ่งหน่วยเช่นเดียวกัน แต่ไม่มีการแลกเปลี่ยน

ข้อมูลอะไรที่เปลี่ยนแปลงไป

```
int investigate(int person1, int person2)
```

- ฟังก์ชันนี้จะถูกเรียกใช้งาน Q ครั้ง
- สำหรับทุกครั้งที่ฟังก์ชันนี้ถูกเรียก จะต้องคืนค่าเวลาต่ำสุดที่ `person1` และ `person2` รับทราบซึ่งกันและกันว่าต่างคนต่างทำการโกง หากทั้งคู่ไม่ได้โกง หรือต่างฝ่ายต่างไม่ทราบว่าอีกฝ่ายโกง ให้คืนค่า -1
- การเรียกฟังก์ชันนี้ไม่ทำให้เวลาในค่ายคอมพิวเตอร์เปลี่ยนแปลง

ข้อจำกัด

- `person1` และ `person2` มีค่าระหว่าง 1 ถึง 200 000
- `person1` \neq `person2` สำหรับการเรียกของทั้งสองฟังก์ชัน
- $N \leq 100\,000$
- $Q \leq 300\,000$

ปัญหาย่อย

1. (15 คะแนน) $N \leq 2\,000$, $Q \leq 2\,000$
2. (30 คะแนน) การเรียก `cheat` ทุกครั้งจะถูกเรียกก่อน `investigate` เสมอ
3. (13 คะแนน) จะมีการเรียก `investigate` เพียงครั้งเดียว ($Q = 1$)
4. (12 คะแนน) รับประกันว่า แต่ละคนมีคนที่รับทราบว่าโกงไม่เกิน 10 คน
5. (30 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

เวลา	การเรียกฟังก์ชัน	คำอธิบาย
1	<code>cheat(1, 2)</code>	1 ทราบว่า 2 โกง และ 2 ก็รับทราบว่า 1 โกง
2	<code>cheat(2, 1)</code>	ไม่เกิดอะไรขึ้น
3	<code>cheat(3, 1)</code>	3 ทราบว่า 1 และ 2 โกง และทั้งสองก็รับทราบว่า 3 โกง
4	<code>cheat(4, 100)</code>	4 ทราบว่า 100 โกง และ 100 ก็รับทราบว่า 4 โกง
	<code>investigate(3, 2)</code>	คืนค่า 3 เพราะ 3 ฟังได้พูดคุยกับ 1 ในเวลาหน่วยที่ 3 และ 1 ได้พูดคุยกับ 2 อยู่แล้ว
	<code>investigate(1, 100)</code>	คืนค่า -1 เพราะทั้งคู่ไม่ทราบข้อมูลของกันและกัน
5	<code>cheat(3, 4)</code>	3 พูดคุยแลกเปลี่ยนกับ 4 ทำให้ 1, 2, 3, 4, และ 100 ทราบว่าต่างโกง
	<code>investigate(1, 100)</code>	คืนค่า 5 เพราะ 1 ทราบว่า 100 โกงในเวลาหน่วยที่ 5
	<code>investigate(1, 2)</code>	คืนค่า 1 เพราะทั้งคู่ได้พูดคุยกัน ตั้งแต่เวลาหน่วยที่ 1

เกรตเดอร์ตัวอย่าง

เกรตเดอร์ตัวอย่างอ่านข้อมูลนำเข้าในรูปแบบต่อไปนี้:

- บรรทัดที่ 1: $N \ Q$
- บรรทัดที่ $1 + i$ (สำหรับ i เมื่อ $1 \leq i \leq N + Q$): C person1 person2 หรือ I person1 person2 แทนการเรียกใช้ฟังก์ชัน `cheat` และ `investigate` ตามลำดับ

เกรตเดอร์ตัวอย่างระบุ Q บรรทัด ซึ่งจะระบุค่าที่คืนมาจากฟังก์ชัน `investigate` ตามลำดับ

Limits

- Time limit: 1.5sec
- Memory limit: 512MB