



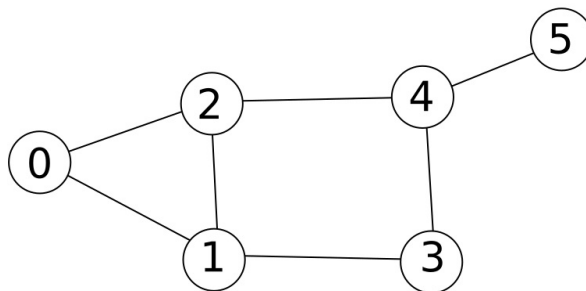
หาบ้าน (findhome)

สุนัขมีสัญชาตญาณดีเยี่ยม เสียดยแค่มันไม่สามารถพูดอะไรได้ หมู่บ้านหนึ่งมีบ้านทั้งสิ้น N หลัง ($N \leq 300$) เรียกเป็นบ้านที่ 0 ถึงบ้านที่ $N - 1$ บ้านเหล่านี้ เชื่อมกันด้วยถนนจำนวน M เส้น ถนนแต่ละเส้นเชื่อมระหว่างบ้านสองหลังเท่านั้น บ้านสองหลังใด ๆ สามารถเดินทางถึงกันได้โดยผ่านถนนเหล่านี้ การเดินทางผ่านถนนหนึ่งเส้นใช้เวลา 1 หน่วย

คุณพบสุนัขหลงทางตัวหนึ่งหลุดออกมาจากบ้านสักหลังในหมู่บ้าน อย่างไรก็ตามมันบอกไม่ได้ว่าบ้านของมันอยู่ที่ไหน แต่ถ้าคุณพามันไปที่บ้านหลังใดหลังหนึ่ง สมมติว่าเป็นบ้านที่ u ถ้านั่นไม่ใช่บ้านของมัน มันจะสามารถบอกได้ว่าถนนออกจากบ้านหลังนี้เส้นใดอยู่บนเส้นทางที่สั้นที่สุดจากบ้านที่ u ไปยังบ้านของสุนัข (โดยวิ่งออกไปยังหน้าถนน)

คุณสามารถถามได้ไม่เกิน Q ครั้ง

พิจารณาตัวอย่างต่อไปนี้ที่ $N = 6$ และ $M = 7$



สมมติว่าบ้านของสุนัขคือบ้านที่ 3

- ถ้าเราพาสุนัขไปที่บ้านที่ 0 สุนัขจะตอบว่าถนนเส้นที่ออกจากบ้านที่ 0 ที่อยู่บนเส้นทางที่สั้นที่สุดไปยังบ้านของมัน คือถนนไปยังบ้านที่ 1
- ถ้าเราพาสุนัขไปที่บ้านที่ 5 สุนัขจะตอบบ้านที่ 4
- ถ้าเราพาสุนัขไปที่บ้านที่ 2 สุนัขจะตอบบ้านที่ 1 (หรืออาจจะตอบบ้านที่ 4 ก็ได้เช่นเดียวกัน)
- ถ้าเราพาสุนัขไปที่บ้านที่ 3 สุนัขจะตอบบ้านที่ 3

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

```
int find_home(int N, int M, vector<pair<int,int>> R)
```

- ฟังก์ชันนี้จะถูกเรียกใช้หนึ่งครั้ง

- อาร์เรย์ R ระบุข้อมูลถนนจำนวน M เส้น กล่าวคือสำหรับ i ที่ $0 \leq i < M$ ถนน i เชื่อมระหว่างบ้าน $R[i].first$ กับ $R[i].second$
- ฟังก์ชันจะต้องคืนค่าเป็นหมายเลขบ้านของสุนัข

ฟังก์ชันดังกล่าวสามารถเรียกใช้ฟังก์ชันต่อไปนี้เพื่อพาสุนัขไปทดสอบ

```
int check(int u)
```

- คุณจะต้องส่งค่า u แทนหมายเลขบ้านที่ต้องการทดสอบ
- ฟังก์ชันจะคืนค่าเป็นหมายเลขบ้าน ถ้าบ้าน u เป็นบ้านของสุนัขจะคืนค่า u กลับมา ถ้าไม่ใช่ จะคืนค่าหมายเลขบ้าน v โดยที่รับประกันว่ามีถนนเชื่อมระหว่างบ้าน u และ v และจะมีเส้นทางที่สั้นที่สุดจากบ้าน u ไปหาบ้านของสุนัขที่ผ่านบ้าน v

เงื่อนไข

- $2 \leq N \leq 300$
- $2 \leq M \leq 45\,000$
- $0 \leq R[i].first \leq N - 1, 0 \leq R[i].second \leq N - 1$ สำหรับ $0 \leq i < M$
- $8 \leq Q \leq 300$

ปัญหาย่อย

1. (2 points) $Q = 300$
2. (5 points) $Q = 50$ และ ทุก ๆ บ้านมีเส้นทางสั้นที่สุดไปยังบ้านของสุนัขที่ใช้ถนนไม่เกิน 49 เส้น
3. (9 points) $Q = 9, M = N - 1$, และ ถนนเชื่อมต่อเป็นเส้น กล่าวคือมีบ้าน 2 หลังที่เชื่อมกับถนนเส้นเดียว ขณะที่บ้านที่เหลือทั้งหมดเชื่อมกับถนนสองเส้น
4. (13 points) $Q = 9, M = N$, และ ถนนเชื่อมต่อเป็นวง กล่าวคือบ้านทุกหลังเชื่อมกับถนนสองเส้น
5. (24 points) $Q = 9, M = N - 1$
6. (13 points) $Q = 20$
7. (34 points) $Q = 9$

ตัวอย่าง

จากตัวอย่างข้างต้น เกรดเดอร์ตัวอย่างจะเรียกใช้ฟังก์ชัน `find_home` ดังนี้

```
find_home(6, 7,
          [[0,2], [0,1], [2,1], [4,2], [1,3], [4,3], [4,5]])
```

ฟังก์ชันดังกล่าว อาจจะเรียกฟังก์ชัน `check` ดังนี้

```
check(0)
```

ซึ่งจะคืนค่า 1

```
check(5)
```

ซึ่งจะคืนค่า 4

```
check(2)
```

ซึ่งจะคืนค่า 1

```
check(3)
```

ซึ่งจะคืนค่า 3

ฟังก์ชัน `find_home` ที่ทำงานถูกต้องจะต้องคืนค่า 3

เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างอ่านข้อมูลนำเข้าในรูปแบบดังนี้

- บรรทัดที่ 1: $N \ M \ Q \ H$
- บรรทัดที่ $2 + i$ ถึง $2 + M - 1$: $R[i].first \ R[i].second$ สำหรับ $0 \leq i < M$
- บรรทัดที่ $2 + M$: $C[0] \ C[1] \ C[2] \ \dots \ C[N - 1]$

มีรายละเอียดการทำงานดังนี้

- บ้านสุนัขคือบ้านที่ H
- เมื่อมีการเรียก `check(u)` ฟังก์ชันจะคืนค่า $C[u]$
- ถ้าเรียก `check` มากกว่า Q ครั้ง เกรตเตอร์จะฟ้องความผิดพลาด
- เมื่อฟังก์ชัน `find_home` จบการทำงาน เกรตเตอร์จะตรวจสอบผลลัพธ์และพิมพ์จำนวนคำถามที่ใช้

ขีดจำกัด

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