



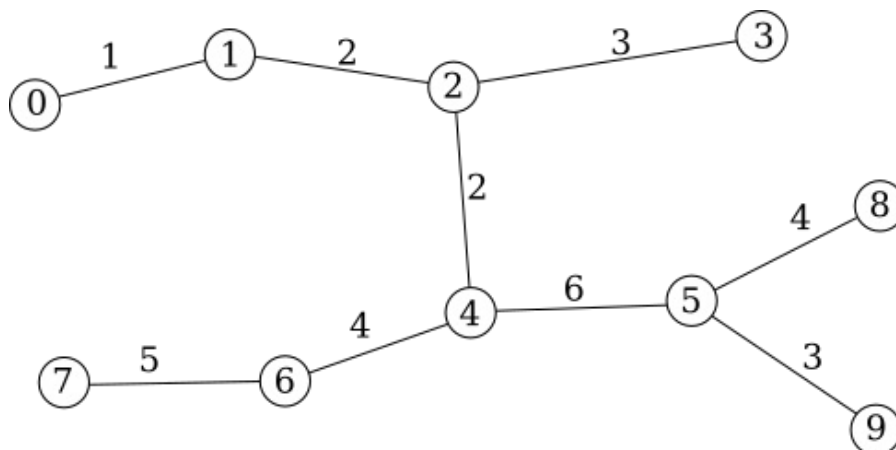
## นัดพบ

ประเทศหนึ่งมีเมืองทั้งสิ้น  $N$  เมือง (เรียกเป็นเมืองหมายเลข 0 ถึงหมายเลข  $N - 1$ ) มีถนนเชื่อมเมืองเหล่านี้จำนวน  $N - 1$  เส้น ถนนแต่ละเส้นจะเชื่อมระหว่างเมืองสองเมือง โดยรับประกันว่าทุกเมืองสามารถเดินทางไปถึงกันได้ผ่านทางถนนเหล่านี้ ถนนแต่ละเส้นมีความยาวระบุไว้ การเดินทางบนถนนเส้นดังกล่าวจะใช้เวลาเท่ากับควมยาวของถนน

มีเพื่อนรักอยู่สามคน คือ  $A$   $B$  และ  $C$  ต้องการจะนัดเจอกัน ในวันที่จะเจอกันนั้น  $A$  อยู่ที่เมือง  $a$ ,  $B$  อยู่ที่เมือง  $b$  และ  $C$  อยู่ที่เมือง  $c$  พวกเขาต้องการเลือกเมือง  $w$  เพื่อนัดเจอกัน เนื่องจากทั้งสามไม่ต้องการให้มีใครเดินทางมากกว่าคนอื่น ๆ มากนัก พวกเขาต้องการเลือกเมือง  $w$  ให้ระยะทางที่มากที่สุดแต่ละคนต้องเดินทางจากเมืองเริ่มต้นไปยังเมือง  $w$  มีค่าน้อยที่สุด นั่นคือ ถ้าให้  $d(u, v)$  แทนระยะทางระหว่างเมือง  $u$  ไปยังเมือง  $v$  เราต้องการเลือกเมือง  $x$  ให้ค่า  $\max[d(a, x), d(b, x), d(c, x)]$  มีค่าน้อยที่สุด

เนื่องจากทั้งสามเป็นเพื่อนรักกันมาก จึงจะนัดกันทั้งสิ้น  $Q$  ครั้ง

พิจารณาตัวอย่างต่อไปนี้ ด้านล่างแสดงตัวอย่างประเทศที่  $N = 10$



สมมติว่าในการนัดครั้งหนึ่ง  $A$  อยู่ที่เมือง 0,  $B$  อยู่ที่เมือง 3 และ  $C$  อยู่ที่เมือง 6 ถ้าทั้งสามตกลงว่าจะไปเจอกันที่เมือง 6 จะได้ว่า  $A$  และ  $B$  ต้องเดินทาง 9 หน่วย ส่วน  $C$  ไม่ต้องเดินทางเลย ทำให้ระยะทางมากที่สุดของทั้งสามคนคือ 9 หน่วย อย่างไรก็ตามถ้าเลือกไปเจอกันที่เมือง 4 แล้วระยะทางมากที่สุดที่ทั้งสามคนต้องเดินทางคือ 5 และนี่คือระยะทางมากที่สุดของทั้งสามคนที่น้อยที่สุดที่เป็นไปได้ ดังนั้นในคำถามนี้ จะต้องตอบ 5

พิจารณาอีกกรณีที่มีการนัดกันที่  $A$  อยู่ที่เมือง 0,  $B$  อยู่ที่เมือง 3 และ  $C$  อยู่ที่เมือง 1 สังเกตว่า ถ้าไปนัดเจอกันที่เมือง 2 จะมีการเดินทางที่มากที่สุดน้อยที่สุดคือ 3 หน่วย

## รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

```
vector<int> best_meeting(int N, int Q,
                        vector<vector<int>>& roads,
                        vector<int>& lengths,
                        vector<int>& a,
                        vector<int>& b,
                        vector<int>& c)
```

- เวกเตอร์ *roads* และ *length* จะมีขนาด  $N - 1$  โดยที่สำหรับ  $0 \leq i \leq N - 2$  ถนนเส้นที่  $i$  จะเชื่อมระหว่างเมือง  $roads[i][0]$  กับ  $roads[i][1]$  และมีความยาว  $lengths[i]$
- เวกเตอร์ *a* *b* และ *c* จะมีขนาด  $Q$  แทนคำถามทั้งหมด สำหรับ  $0 \leq j \leq Q - 1$  ในคำถามที่  $j$  เมืองของทั้งสามคนคือ  $a[j]$   $b[j]$  และ  $c[j]$
- ฟังก์ชันจะต้องคืนค่าเป็นเวกเตอร์ที่มีขนาด  $Q$  เวกเตอร์ในช่องที่  $j$  จะเป็นคำตอบของคำถามที่  $j$  นั่นคือ เป็นระยะทางที่ทั้งสามคนต้องเดินทางที่มากที่สุด在三คนเพื่อไปยังเมือง  $w$  ที่มีค่าน้อยที่สุดที่เป็นได้

## ขอบเขต

- $1 \leq N \leq 100\,000, 1 \leq Q \leq 100\,000$
- สำหรับ  $0 \leq i < N - 1, \quad 0 \leq roads[i][0] \leq N - 1, \quad 0 \leq roads[i][1] \leq N - 1, \quad 1 \leq lengths[i] \leq 20\,000.$
- สำหรับ  $0 \leq j \leq Q - 1, 0 \leq a[j] \leq N - 1, 0 \leq b[j] \leq N - 1, 0 \leq c[j] \leq N - 1.$

## ปัญหาย่อย

1. (2 คะแนน)  $N, Q \leq 1\,000$
2. (3 คะแนน)  $N \leq 1\,000, Q \leq 50\,000$
3. (5 คะแนน) ถนนในประเทศนี้มีลักษณะเป็นเส้นตรง นั่นคือ ถนนเส้นที่  $i$  จะเชื่อมระหว่างเมือง  $i$  กับ  $i + 1$
4. (7 คะแนน)  $a[j] = b[j]$  เสมอ
5. (8 คะแนน) เส้นทางระหว่างเมือง  $a[j]$  ไปยัง  $b[j]$  จะมีถนนไม่เกินสองเส้น
6. (25 คะแนน)  $N, Q \leq 50\,000$
7. (50 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

## ตัวอย่าง

จากตัวอย่างข้างต้น การเรียกฟังก์ชัน `best_meeting` จะเป็นดังนี้

```
best_meeting(10, 2,
              [[0,1], [1,2], [3,2], [2,4], [7,6],
               [4,6], [4,5], [5,8], [5,9]],
              [1, 2, 3, 2, 5, 4, 6, 4, 3],
              [0, 0],
              [3, 3],
              [6, 1])
```

ฟังก์ชันจะต้องคืนค่า [5, 3]

## เกรตเดอร์ตัวอย่าง

เกรตเดอร์ตัวอย่างอ่านข้อมูลดังนี้

- บรรทัดที่ 1:  $N, Q$
- บรรทัดที่  $2, \dots, N$ :  $roads[i][0], roads[i][1], length[i]$
- บรรทัดที่  $N + 1, \dots, N + Q$ :  $a[j], b[j], c[j]$

เกรตเดอร์จะพิมพ์ค่าที่คืนจากฟังก์ชัน `best_meeting` บรรทัดละตัว

## ข้อจำกัด

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