แพะจ่าย (Goat Pay)

แพะและมะนาวอยู่บนต้นไม้ต้นหนึ่งที่มีจุดยอด N จุด สำหรับเส้นเชื่อมที่ i ที่เชื่อมระหว่างจุดยอด U_i และ V_i ของต้นไม้นี้จะมีสี C_i และ มีระยะห่างระหว่างทั้งสองจุดยอดเท่ากับ W_i

เนื่องจากต้นไม้นี้เป็นต้นไม้วิเศษที่ถูกปลูกและดูแลโดยเทพเจ้าสุริยันแสงจ้าทำให้การเดินทางบนต้นไม้นี้จากจุดยอด u ไปยัง v จะใช้เสียค่าเดินทางเท่ากับ การ XOR ของระยะทางทั้งหมดบนเส้นทางจากจุดยอด u ไปยัง v

ไม่เพียงแค่นั้นบางครั้งเทพเจ้าสุริยันแสงจ้าก็อยากจะเปลี่ยนระยะห่างของเส้นเชื่อมที่ id ให้กลายเป็น d ด้วยเพื่อไม่ให้แพะและมะนาวจำค่าใช้จ่ายในการเดินไปหากันได้

ในเวลาทั้งหมด Q วัน เทพเจ้าสุริยันแสงจ้าจะทำอยู่สองอย่าง ได้แก่

1. ถาม**คุณ**ว่าหากแพะและมะนาวอยู่ที่จุดยอด u และ v ตามลำดับ หากพิจารณาหากเทพเจ้สสุริยันแสงจ้าจะคิดราคาค่าเดินทางเพียงแค่เส้นเชื่อมที่มีสี x เท่านั้น ถ้าแพะจะเดินทางไปหามะนาวต้องจ่ายเงินทั้งหมดเท่าใด
2. เปลี่ยนระยะห่างของเส้นเชื่อม id ให้กลายเป็น d

ข้อมูลที่อาจเป็นประโยชน์

XOR ($\oplus$) เป็นการดำเนินการบนระบบเลขฐานสองโดยการเปรียบเทียบบิตของสองจำนวน โดยเปรียบเทียบทีละบิตและให้ผลลัพธ์ตามกฎของ XOR:

- ถ้าบิตที่เปรียบเทียบ เหมือนกัน (ทั้งคู่เป็น 0 หรือทั้งคู่เป็น 1) ผลลัพธ์จะเป็น 0
- ถ้าบิตที่เปรียบเทียบ ต่างกัน (บิตหนึ่งเป็น 0 และอีกบิตเป็น 1) ผลลัพธ์จะเป็น 1

ตัวอย่างเช่น

- $9 \oplus 4 = 1001_2 \oplus 0100_2 = 1101_2 = 13$
- $5 \oplus 1 = 101_2 \oplus 001_2 = 100_2 = 4$

การดำเนินการ XOR ในภาษา C/C++ สามารถใช้ตัวดำเนินการ ^ ได้

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันดังต่อไปนี้

```
void plant_tree(int N, vector<int> U, vector<int> V, vector<int> W, v
ector<int> C)
```

- ฟังก์ชันนี้จะถูกเรียกเพียงครั้งเดียว ก่อนการเรียกฟังก์ชัน goat_pay และ update_edge ทั้งหมด
- ตัวแปร N ระบุจำนวนจุดยอดของต้นไม้
- ตัวแปร U, V, W และ C ระบุการมีเส้นเชื่อมระหว่างจุดยอด U[i] และ V[i] โดยที่ระยะห่างระหว่างทั้งสองจุดยอดเท่ากับ W[i] และมีสี C[i]

```
int goat_pay(int u, int v, int x)
```

- ฟังก์ชันนี้จะถูกเรียกหลายครั้ง โดยจำนวนการเรียกฟังก์ชันนี้รวมกับฟังก์ชัน update_edge จะเท่ากับ Q ครั้งพอดี
- ตัวแปร u ระบุตำแหน่งของแพะ
- ตัวแปร v ระบุตำแหน่งของมะนาว
- ตัวแปร x ระบุสีที่เทพเจ้าสุริยันแสงจ้าเลือก
- ฟังก์ชันนี้จะคืนค่า จำนวนเงินที่แพะต้องจ่ายเพื่อที่จะเดินทางไปหามะนาว

```
void update_edge(int id, int d)
```

- ฟังก์ชันนี้จะถูกเรียกหลายครั้ง โดยจำนวนการเรียกฟังก์ชันนี้รวมกับฟังก์ชัน find_dist จะเท่ากับ Q ครั้งพอดี
- ตัวแปร id ระบุหมายเลขของเส้นเชื่อมที่ถูกเปลี่ยนระยะทาง
- ตัวแปร d ระบุระยะทางใหม่ของเส้นเชื่อมนั้น

ข้อจำกัด

- $2 \leq N \leq 1\,000\,000$
- $1 \leq Q \leq 1\,000\,000$
- $0 \leq U_i, V_i \leq N-1$ และ $U_i \neq V_i$ สำหรับ $i = 0, 1, 2, \ldots, N-2$
- $0 \leq W_i \leq 10^8$ สำหรับ $i = 0, 1, 2, \ldots, N-2$
- $1 \leq C_i \leq N$ สำหรับ $i = 0, 1, 2, \ldots, N-2$
- $0 \leq u_i, v_i \leq N-1$ สำหรับ $i = 0, 1, 2, \ldots, Q-1$
- $1 \leq x_i \leq N$ สำหรับ $i = 0, 1, 2, \ldots, Q-1$
- $0 \leq id_i \leq N-2$ สำหรับ $i = 0, 1, 2, \ldots, Q-1$
- $1 \leq d_i \leq 10^8$ สำหรับ $i = 0, 1, 2, \ldots, Q-1$

ปัญหาย่อย

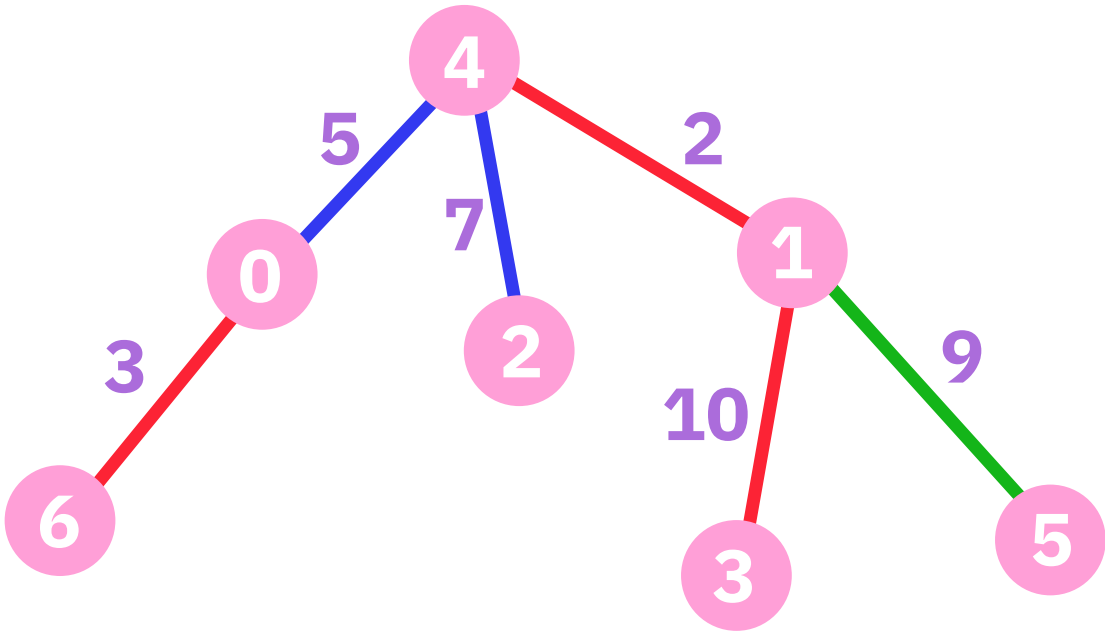
1. (5 คะแนน) $U_i = i, V_i = i+1, C_i = 1$ และ $x_i = 1$
2. (4 คะแนน) $U_i = i, V_i = i+1, C_i \leq 20$ และ $x_i \leq 20$
3. (8 คะแนน) $U_i = i$ และ $V_i = i+1$
4. (17 คะแนน) $N \leq 1\,000$ และ $Q \leq 2\,000$
5. (1 คะแนน) ไม่มีการเรียกฟังก์ชัน update_edge และ $W_i = 0$

6. (8 คะแนน) ไม่มีการเรียกฟังก์ชัน update_edge, $C_i = 1$ และ $x_i = 1$
7. (10 คะแนน) ไม่มีการเรียกฟังก์ชัน update_edge
8. (5 คะแนน) $N \leq 100\,000, Q \leq 100\,000, C_i = 1$ และ $x_i = 1$
9. (6 คะแนน) $N \leq 100\,000, Q \leq 100\,000, C_i \leq 20$ และ $x_i \leq 20$
10. (9 คะแนน) $N \leq 100\,000$ และ $Q \leq 100\,000$
11. (7 คะแนน) $C_i = 1$ และ $x_i = 1$
12. (7 คะแนน) $C_i \leq 20$ และ $x_i \leq 20$
13. (13 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

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

การเรียกฟังก์ชันนี้จะได้ต้นไม้



เมื่อเส้นเชื่อมสีแดงคือสีหมายเลข 1 เส้นเชื่อมสีน้ำเงินหมายถึงสีหมายเลข 2 และเส้นเชื่อมสีเขียวคือสีหมายเลข 3

ถัดมามีการเรียกฟังก์ชัน goat_pay

```
goat_pay(6, 3, 1)
```

แพะจะทำการคำนวณจำนวนเงินที่ต้องใช้เฉพาะส่วนที่ผ่านหมายเลข 1 ซึ่งจะได้เท่ากับ $3 \oplus 2 \oplus 10 = 11$

ดังนั้นฟังก์ชัน goat_pay จะคืนค่า 11

ถัดมามีการเรียกฟังก์ชัน `goat_pay`

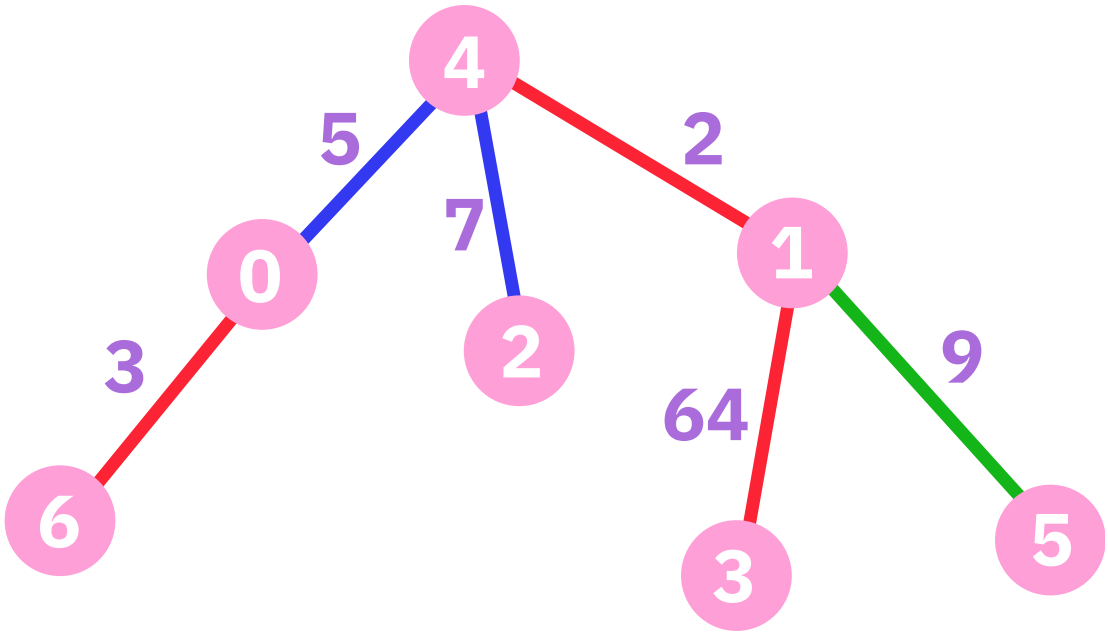
```
goat_pay(6, 3, 2)
```

แพะจะทำการคำนวณจำนวนเงินที่ต้องใช้เฉพาะส่วนที่ผ่านหมายเลข 2 ซึ่งจะได้เท่ากับ 5 ดังนั้นฟังก์ชัน `goat_pay` จะคืนค่า 5

ถัดมามีการเรียกฟังก์ชัน `update_edge`

```
update_edge(2, 64)
```

ทำการเปลี่ยนระยะห่างของเส้นเชื่อมหมายเลข 2 ซึ่งเชื่อมระหว่างจุดยอด 3 และ 1 ให้กลายเป็น 64




```
goat_pay(6, 3, 1)
```

แพะจะทำการคำนวณจำนวนเงินที่ต้องใช้เฉพาะส่วนที่ผ่านหมายเลข 1 ซึ่งจะได้เท่ากับ $3 \oplus 2 \oplus 64 = 65$

ดังนั้นฟังก์ชัน `goat_pay` จะคืนค่า 65

เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างจะอ่านข้อมูลดังนี้:

- บรรทัดที่ 1: N Q
- บรรทัดที่ 2 ถึง N: U_i V_i W_i C_i สำหรับ $i = 0, 1, 2, \ldots N-2$

- บรรทัดที่ $N+1$ ถึง $N+Q$: แบ่งออกเป็น 2 รูปแบบ ได้แก่

1. $1 \;\; u \;\; v \;\; x$ สำหรับการถามฟังก์ชัน `goat_pay`
2. $2 \;\; id \;\; d$ สำหรับการถามฟังก์ชัน `update_edge`

เกรดเดอร์ตัวอย่างจะส่งออกค่าที่ได้รับจากฟังก์ชัน `goat_pay`

ขอบเขต

- Time limit: 4.5 seconds
- Memory limit: 512 MB