

ระบายสีต้นไม้

อลิซและบ๊อบกำลังเล่นเกมกับต้นไม้ต้นหนึ่ง ต้นไม้นี้มีจุดยอด N จุด หมายเลขตั้งแต่ 0 ถึง $N - 1$ และมีเส้นเชื่อม $N - 1$ เส้น

ระยะทางระหว่างจุดยอดสองจุด u และ v คือจำนวนเส้นเชื่อมบนเส้นทางเดียวในต้นไม้ที่เชื่อมจาก u ไปยัง v เราจะเขียนระยะทางนี้เป็น $\text{dist}(u, v)$

เกมนี้แบ่งเป็นสองขั้นตอน

ในขั้นตอนแรก อลิซจะเห็นต้นไม้ทั้งต้น อลิซต้องกำหนด สี ให้กับจุดยอดทุกจุด โดยสีของจุดยอดหนึ่งจุดคือสตริงฐานสองหนึ่งสตริง กล่าวคือเป็นสตริงที่ประกอบด้วยอักขระ 0 และ 1 เท่านั้น

หลังจากนั้น ต้นไม้จะถูกซ่อนไว้ทั้งหมด ในแต่ละคำถาม เกรตเดอร์จะเลือกจุดยอดสองจุด u และ v ซึ่งอาจเป็นจุดยอดเดียวกันก็ได้ แล้วบ๊อบจะได้รับเพียงสีของจุดยอดสองจุดนั้น บ๊อบจะไม่ทราบหมายเลขของจุดยอดทั้งสอง ไม่เห็นต้นไม้ (ไม่ทราบค่า N ด้วย) และไม่เห็นสีของจุดยอดอื่น ๆ บ๊อบต้องตอบค่า $\text{dist}(u, v)$ ให้ถูกต้อง

อลิซและบ๊อบสามารถตกลงวิธีการระบายสีและวิธีการถอดรหัสไว้ล่วงหน้าได้ แต่เมื่อบ๊อบตอบคำถามหนึ่งคำถาม บ๊อบจะใช้ข้อมูลได้เฉพาะสีสองสีที่ได้รับในคำถามนั้น คำถามแต่ละคำถามเป็นอิสระจากกัน

คุณต้องเขียนโปรแกรมของอลิซและบ๊อบ โดยต้องทำให้ทุกส้อมีความยาวไม่เกินขอบเขตที่กำหนด และบ๊อบตอบระยะทางได้ถูกต้องทุกคำถาม

รายละเอียดการเขียนโปรแกรม

คุณต้องเขียนฟังก์ชันต่อไปนี้

```
vector<string> Alice(int N, vector<int> U, vector<int> V)
```

- ฟังก์ชันนี้จะถูกเรียกหนึ่งครั้งสำหรับต้นไม้หนึ่งต้น
- N คือจำนวนจุดยอดของต้นไม้
- U และ V เป็นอาร์เรย์ความยาว $N - 1$ ที่ระบุเส้นเชื่อมของต้นไม้ สำหรับ $0 \leq i < N - 1$ จะมีเส้นเชื่อมระหว่างจุดยอด $U[i]$ และ $V[i]$
- ฟังก์ชันต้องคืนอาร์เรย์ C ความยาว N โดย $C[x]$ คือสีของจุดยอด x
- ทุกสีต้องเป็นสตริงฐานสอง กล่าวคือมีเฉพาะอักขระ 0 และ 1
- ความยาวของสีทุกสีต้องไม่เกิน $B(N)$ ตามที่ระบุในหัวข้อการให้คะแนน

```
int Bob(string X, string Y)
```

- ฟังก์ชันนี้จะถูกเรียกหนึ่งครั้งต่อหนึ่งคำถาม
- X และ Y คือสีของจุดยอดสองจุดในต้นไม้เดียวกัน จุดยอดสองจุดนี้อาจเป็นจุดยอดเดียวกันก็ได้
- ฟังก์ชันต้องคืนระยะทางระหว่างจุดยอดสองจุดนั้น
- ฟังก์ชัน Bob ต้องไม่ใช่ข้อมูลที่ถูกคำนวณหรือบันทึกไว้โดยฟังก์ชัน Alice ผ่านตัวแปร global หรือหน่วยความจำอื่นใด

ในการตรวจ คำตอบของคุณอาจถูกทดสอบกับต้นไม้หลายต้น เกรตเดอร์อาจเรียก Alice สำหรับต้นไม้หนึ่งต้น แล้วนำสีที่ได้ไปใช้เรียก Bob หลายครั้ง โปรแกรมของ Bob ต้องสามารถตอบได้จากสตริงสองสตริงที่ได้รับเท่านั้น

นอกจากนี้ โปรแกรม Bob จะไม่สามารถเก็บข้อมูลใดๆ ที่เป็นตัวแปร global หรือ static ข้ามระหว่างคำถามได้ เกรตเดอร์จริงอาจจะเรียกฟังก์ชัน Bob หลายๆ ครั้งโดยสลับไปมาระหว่างต้นไม้หลายต้น ดังนั้น คุณจะต้องตรวจสอบว่าฟังก์ชันที่คุณเขียน ไม่ได้มีการใช้งานตัวแปร global หรือ static หรือหน่วยความจำอื่น ๆ ในการพยายามเก็บข้อมูลข้ามระหว่างคำถาม

เงื่อนไข

- $2 \leq N \leq 1000$
- $0 \leq U[i], V[i] < N$ สำหรับทุก $0 \leq i < N - 1$
- เส้นเชื่อมที่กำหนดให้จะประกอบกันเป็นต้นไม้

ปัญหาย่อย

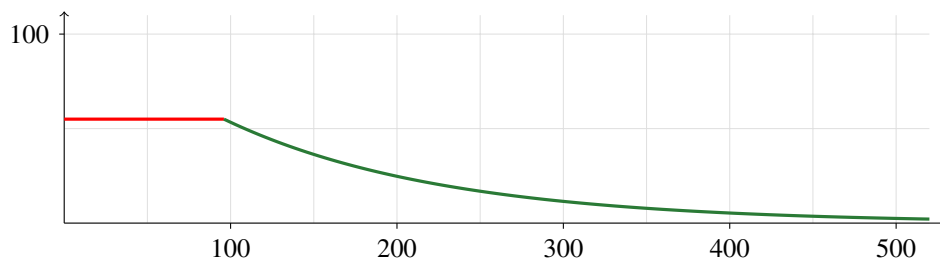
ปัญหาย่อย	คะแนน	เงื่อนไขเพิ่มเติม	ขอบเขต B
1	8	$N \leq 200$	1600
2	7	ต้นไม้เป็นเส้นทาง กล่าวคือจุดยอดทุกจุดมีดีกรีไม่เกิน 2	350
3	10	ต้นไม้มีจุดยอดหนึ่งจุดที่อยู่ห่างจากทุกจุดยอดไม่เกิน 2	350
4	10	มีจุดยอดไม่เกินหนึ่งจุดที่มีดีกรีมากกว่า 2	350
5	10	ต้นไม้เป็นต้นไม้ไบนารีสมบูรณ์ กล่าวคือมีรากหนึ่งจุด ทุกจุดยอดที่ไม่ใช่ใบมีลูกสองจุด และใบทุกใบอยู่ที่ความลึกเดียวกัน	350
6	55	ไม่มีเงื่อนไขเพิ่มเติม	1005

คะแนนของปัญหาย่อยที่ 1–5 พิจารณาดังนี้ ให้ B' เป็นความยาวของสีที่ยาวที่สุดที่อลิซระบายในปัญหาย่อยนั้น แล้วจะได้คะแนนทั้งหมดของปัญหาย่อยหาก $B' \leq B$

คะแนนของปัญหาย่อยที่ 6 พิจารณาดังนี้ ให้ B' เป็นความยาวของสีที่ยาวที่สุดที่อลิซระบาย แล้วคะแนน F ที่ได้ของปัญหาย่อยที่ 6 นั้น คำนวณโดย

$$F = \begin{cases} 55, & B' \leq 96, \\ 55 \cdot \exp(-0.00767528(B' - 96)), & B' > 96. \end{cases}$$

และกราฟของฟังก์ชัน F ที่ขึ้นกับ B' แสดงด้านล่าง



การให้คะแนน

ถ้าสีใดสีหนึ่งที่ Alice ระบายมีความยาวมากกว่าขอบเขต B ของปัญหาย่อยนั้น หรือมีอักขระที่ไม่ใช่ 0 หรือ 1 จะถือว่าคำตอบผิด

ถ้าในการทดสอบใด ๆ มีคำถามที่ Bob ตอบระยะทางผิด จะถือว่าคำตอบผิดสำหรับปัญหาย่อยนั้น

คะแนนของแต่ละปัญหาย่อยจะได้ก็ต่อเมื่อโปรแกรมผ่านการทดสอบในปัญหาย่อยนั้น

ตัวอย่าง

พิจารณาด้านไม้ที่มี $N = 5$ และมีเส้นเชื่อมดังนี้

```
0 1
1 2
1 3
3 4
```

ต้นไม้นี้มีระยะทางบางค่าดังนี้

- $\text{dist}(0, 2) = 2$
- $\text{dist}(2, 4) = 3$
- $\text{dist}(3, 4) = 1$

สมมติว่า Alice คินสีดังนี้

```
C[0] = "000"
C[1] = "001"
C[2] = "010"
C[3] = "011"
C[4] = "100"
```

สีชุดนี้เป็นเพียงตัวอย่างรูปแบบการคินค่าเท่านั้น และโดยทั่วไปไม่เพียงพอสำหรับการตอบระยะทางของต้นไม้ทุกต้น

ถ้าเกรดเดอร์ต้องการถามระยะทางระหว่างจุดยอด 2 และ 4 เกรดเดอร์จะเรียก

```
Bob("010", "100")
```

ฟังก์ชัน Bob ต้องคืนค่า 3

เกรดเดอร์ตัวอย่าง

คุณจะได้รับไฟล์ treecoloring.h และ sample_grader.cpp เพื่อใช้ทดสอบในเครื่องของคุณ

ไฟล์ treecoloring.h ประกาศฟังก์ชันที่คุณต้องเขียนดังนี้

```
vector<string> Alice(int N, vector<int> U, vector<int> V);
int Bob(string X, string Y);
```

คุณควรเขียนคำตอบของคุณในไฟล์ เช่น treecoloring.cpp แล้วคอมไพล์ร่วมกับเกรดเดอร์ตัวอย่าง

```
g++ -std=c++17 -O2 -pipe -static -s sample_grader.cpp treecoloring.cpp -o treecoloring
```

เกรดเดอร์ตัวอย่างอ่านข้อมูลจาก standard input ในรูปแบบต่อไปนี้

```
T
for each testcase:
  N Q B
  U[0] V[0]
  U[1] V[1]
  ...
  U[N-2] V[N-2]
  A[0] C[0]
  A[1] C[1]
  ...
  A[Q-1] C[Q-1]
```

ความหมายของอินพุตมีดังนี้

- T คือจำนวน testcase ในไฟล์อินพุต
- สำหรับแต่ละ testcase บรรทัดแรกมีจำนวนเต็ม N Q B โดย N คือจำนวนจุดยอด Q คือจำนวนคำถาม และ B คือขอบเขตความยาวสีที่เกรดเดอร์ตัวอย่างจะใช้ตรวจ
- ถัดมามี $N - 1$ บรรทัด แต่ละบรรทัดมี $U[i]$ $V[i]$ แทนเส้นเชื่อมหนึ่งเส้นของต้นไม้ จุดยอดมีหมายเลขตั้งแต่ 0 ถึง $N - 1$
- ถัดมามี Q บรรทัด แต่ละบรรทัดมี $A[i]$ $C[i]$ แทนจุดยอดสองจุดในคำถามที่ i เกรดเดอร์จะถามระยะทางระหว่างสองจุดนี้

เกรดเดอร์ตัวอย่างจะเรียก Alice หนึ่งครั้ง จากนั้นเรียก Bob สำหรับคำถามทั้งหมดในโปรเซสเดียวกัน ซึ่งแตกต่างจากเกรดเดอร์จริงที่จะถามคำถามในโปรเซสใหม่ทุกครั้ง ทำให้ Bob เก็บค่าในตัวแปรข้ามระหว่างคำถามไม่ได้ นอกจากนี้ เกรดเดอร์ตัวอย่างคำนวณระยะทางทุกคู่เพื่อใช้ตรวจคำตอบ จึงเหมาะสำหรับการทดสอบกับข้อมูลขนาดเล็กเท่านั้น

ขีดจำกัด

- เวลา 5 วินาที
- หน่วยความจำ: 256 MiB

หมายเหตุ: เวลา 5 วินาทีนี้รวมเวลาที่ Alice ใช้และเวลาที่ Bob ใช้ตอบคำถาม Q ครั้งรวมกัน ($Q \leq 100$)