

การแสดงมายากล (Magic Show)

Alice กับ Bob เป็นนักมายากลที่มีชื่อเสียง Catherine ผู้เป็นหญิงร่ำรวยที่สนใจในความสามารถของนักมายากลทั้งสอง ได้ประกาศว่าจะให้รางวัลมูลค่าสูงแก่นักมายากลทั้งสอง ถ้าพวกเขาสามารถแสดงมายากลตามที่ระบุด้านล่างได้ รายละเอียดของมายากลเป็นดังนี้

- ขั้นที่ 1: Bob เข้าไปในห้องที่ถูกตัดขาดจากโลกภายนอกโดยสิ้นเชิง Bob จะสามารถสื่อสารกับ Catherine ได้เท่านั้น หลังจากนั้น Alice จะบอกจำนวน n ที่มีค่าระหว่าง 2 ถึง 5 000 กับ Catherine
- ขั้นที่ 2: Catherine จะบอกจำนวน X กับ Alice โดยที่จำนวนนี้จะมีค่าระหว่าง 1 ถึง 10^{18}
- ขั้นที่ 3: Alice จะสร้างต้นไม้ (tree) ที่มีจำนวนจุดยอดเท่ากับ n จุดพอดี และส่งให้กับ Catherine
- ขั้นที่ 4: Catherine จะลบเส้นเชื่อมไม้เกิน $\lfloor \frac{n-2}{2} \rfloor$ เส้นจากต้นไม้ดังกล่าว และส่งเส้นเชื่อมที่เหลือให้กับ Bob
- ขั้นที่ 5: Bob จะพิจารณารกราฟดังกล่าวอย่างละเอียด และตอบว่าจำนวนที่ Catherine บอกกับ Alice คืออะไร

อย่างไรก็ตาม Alice กับ Bob ไม่คิดว่าพวกเขาจะสามารถเล่นมายากลดังกล่าวได้สำเร็จ จึงได้มาขอความช่วยเหลือจากคุณ ให้คุณช่วยเขียนโปรแกรมที่ implement กลยุทธ์ของ Alice และกลยุทธ์ของ Bob เพื่อที่จะสามารถแก้ปัญหาที่ Catherine เสนอมาได้

รายละเอียดการเขียนโปรแกรม

คุณจะต้องส่งไฟล์เป็นจำนวน **สองไฟล์**:

ไฟล์แรกชื่อ `Alice.cpp` ที่เขียนกลยุทธ์ของ Alice ไฟล์ดังกล่าวจะต้อง include `Alice.h` โดยใช้ `#include` ฟังก์ชันที่ต้องเขียนในไฟล์นี้คือ:

```
std::vector<std::pair<int, int>> Alice();
```

- ในแต่ละกรณีทดสอบ ฟังก์ชันนี้จะถูกเรียกหนึ่งครั้งเท่านั้นในตอนเริ่มต้น
- ฟังก์ชันจะต้องคืนเวกเตอร์ของคู่ลำดับ (pair) ที่แทนเส้นเชื่อมของต้นไม้ที่ Alice สร้างในขั้นตอนที่ 3 ของมายากล
 - หมายเลขของจุดยอดบนต้นไม้จะต้องเริ่มจาก 1
 - คุณจะต้องรับประกันว่าต้นไม้ที่สร้างนั้นถูกต้อง นั่นคือคุณจะต้องคืนเส้นเชื่อมจำนวน $n - 1$ เส้น และจุดยอดทั้งหมดจะต้องเชื่อมต่อกัน

ฟังก์ชัน `Alice()` จะต้องเรียกฟังก์ชันด้านล่าง **หนึ่งครั้งพอดีเท่านั้น**:

```
long long setN(int n);
```

- Alice จะเลือกพารามิเตอร์ n ที่จะส่งให้กับ Catherine ในขั้นตอนที่ 1 ด้วยฟังก์ชันนี้

- ฟังก์ชันจะคืนค่า X ที่ Catherine ส่งให้กับ Alice ในขั้นตอนที่ 2 ของมายากล

ไฟล์ที่สองชื่อ Bob.cpp จะต้องเขียนกลยุทธ์ของ Bob ไฟล์ดังกล่าวจะต้อง include Bob.h โดยใช้ #include ฟังก์ชันที่ต้องเขียนในไฟล์นี้คือ:

```
long long Bob(std::vector<std::pair<int, int>> V);
```

- สำหรับแต่ละกรณีทดสอบ ฟังก์ชันนี้จะถูกเรียกหนึ่งครั้งพอดีหลังจากการเรียกฟังก์ชัน Alice()
- พารามิเตอร์ V จะระบุรายการของเส้นเชื่อมในกราฟที่ Catherine ส่งให้กับ Bob ในขั้นตอนที่ 4 ของการแสดงมายากล
- เส้นเชื่อมที่ส่งให้จะถูกส่งให้ตามลำดับที่ **เรียงลำดับแล้ว** นั่นคือ:
 - สำหรับจุดปลายสองจุดของแต่ละเส้นเชื่อม หมายเลขจุดปลายที่น้อยกว่าจะมาก่อน
 - เส้นเชื่อมทั้งหมดจะถูกเรียงจากน้อยไปหามากโดยเรียงจากจุดปลายแรกก่อน จากนั้นจึงเรียงตามจุดปลายที่สอง
- ฟังก์ชันจะต้องคืนค่าเป็นจำนวนเต็มหนึ่งจำนวนที่แทนจำนวน X

ตัวอย่างการทำงาน

Call	Return Value
Alice()	
setN(4)	3
	{{1, 2}, {2, 3}, {2, 4}}
Bob({{1, 2}, {2, 4}})	3

ตัวอย่างข้างต้นแทนสถานการณ์ต่อไปนี้

- ขั้นที่ 1: ขั้นแรก Alice ส่ง 4 ให้กับ Catherine
- ขั้นที่ 2: Catherine ส่ง 3 ให้กับ Alice
- ขั้นที่ 3: Alice สร้างต้นไม้ที่มี 4 จุดยอด ที่ประกอบด้วยเส้นเชื่อม {{1, 2}, {2, 3}, {2, 4}} และส่งให้กับ Catherine
- ขั้นที่ 4: Catherine ตัดเส้นเชื่อมที่เชื่อมระหว่าง 2 กับ 3 และส่งเส้นเชื่อมที่เหลือ คือ {{1, 2}, {2, 4}} ให้กับ Bob
- ขั้นที่ 5: Bob ตอบจำนวน 3 เนื่องจากคำตอบถูกต้อง จึงถือว่า Alice และ Bob ทำการแสดงมายากลได้สำเร็จ

เงื่อนไข

- $1 \leq X \leq 10^{18}$.

ปัญหาย่อย

- (5 points): $X \leq 5\,000$.
- (30 points): $X \leq 25\,000\,000$.

3. (65 points): ไม่มีข้อไขเพิ่มเติม

เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างจากอ่านข้อมูลนำเข้าในรูปแบบต่อไปนี้:

- บรรทัด 1: T ($T \in \{1, 2\}$)

ถ้า $T = 1$ เกรตเตอร์ตัวอย่างจะอ่านข้อมูลดังต่อไปนี้:

- บรรทัด 2: X ($1 \leq X \leq 10^{18}$)

เกรตเตอร์ตัวอย่างจะพิมพ์คำตอบที่ได้จากฟังก์ชัน `Alice()` ในรูปแบบต่อไปนี้:

- บรรทัด 1: n
- บรรทัด $2 + i$ ($0 \leq i \leq n - 2$): $u[i] \ v[i]$, เมื่อมีเส้นเชื่อมระหว่าง $u[i]$ และ $v[i]$.

ถ้า $T = 2$ เกรตเตอร์ตัวอย่างจะอ่านข้อมูลดังนี้:

- บรรทัด 2: $n \ m$ ($2 \leq n \leq 5000, n - 1 - \lfloor \frac{n-2}{2} \rfloor \leq m \leq n - 1$) โดยที่ n แทนจำนวนจุดยอด และ m แทนจำนวนเส้นเชื่อมที่เหลือ
- บรรทัด $3 + i$ ($0 \leq i \leq m - 1$): $u[i] \ v[i]$ ที่แสดงว่ามีเส้นเชื่อมระหว่าง $u[i]$ และ $v[i]$.

เกรตเตอร์ตัวอย่างจะพิมพ์คำตอบที่ได้จากฟังก์ชัน `Bob()` ในรูปแบบต่อไปนี้:

- บรรทัด 1: X