

MOOZ SERVER

2 seconds, 256 MB

ระบบ Grader ของค่ายโอลิมปิกแห่งหนึ่งประกอบไปด้วยเซิร์ฟเวอร์ N เครื่อง ซึ่งเชื่อมต่อกันผ่านสายเชื่อมต่อ M เส้น หากเซิร์ฟเวอร์ใดต้องการส่งข้อมูลไปยังอีกเซิร์ฟเวอร์หนึ่งจะสามารถส่งข้อมูลกันเป็นทอดทอดผ่านสายเชื่อมต่อบางจำนวนไปยังจุดหมายได้ ทั้งนี้ ไม่มีสายเชื่อมต่อเส้นใดที่ปลายทั้งสองเป็นเซิร์ฟเวอร์เดียวกัน และไม่มีสายเชื่อมต่อสองเส้นคู่ใดที่เชื่อมต่อเซิร์ฟเวอร์คู่เดียวกัน

คุณไม่พึงพอใจกับโจทย์ข้อสอบในวันนี้จึงต้องการที่จะทำลายระบบ Grader ทั้งเสีย โดยจะดึงสายเชื่อมต่อหนึ่งเส้นออกเพื่อทำให้ระบบ Grader ไม่สมบูรณ์ กล่าวคือ คุณต้องการทำให้มีเซิร์ฟเวอร์บางคู่ไม่สามารถติดต่อสื่อสารหากันได้ อย่างไรก็ตาม คุณไม่ทราบว่าจะระบบ Grader นี้มีการเชื่อมต่ออย่างไร แต่โชคเข้าข้างคุณ เพราะคุณสามารถหากลอกถามข้อมูลจากผู้คุมระบบ (ศ.ดร.จิรายุ บัวประทีป) ได้ กล่าวคือ คุณสามารถเลือกเซตของเซิร์ฟเวอร์ S มา (S เป็นสับเซตของ $\{1, 2, 3, \dots, N\}$ และ $1 \leq |S| \leq N-1$) เพื่อสอบถามผู้คุมระบบว่า ถ้าหากเซิร์ฟเวอร์ทั้งหมดที่ไม่อยู่ในเซต M นั้นพังลงและหายไปจากระบบโดยสิ้นเชิง ระบบ Grader จะถูกแตกออกเป็นกี่ระบบย่อย

นิยาม ระบบย่อย คือ ระบบของเซิร์ฟเวอร์บางจำนวนซึ่งทุกเซิร์ฟเวอร์ในระบบย่อยสามารถติดต่อสื่อสารหากันได้ แต่ไม่สามารถติดต่อสื่อสารกับเซิร์ฟเวอร์ใด ๆ นอกกระบบย่อยได้เลย ดังนั้นระบบ Grader ทั้งระบบที่ประกอบไปด้วยเซิร์ฟเวอร์ทั้ง N เครื่อง นับว่าประกอบไปด้วย 1 ระบบย่อย

คุณต้องการที่จะทราบว่า มีสายเชื่อมต่อเส้นใดบ้างที่เมื่อคุณดึงเส้นนั้นออกเพียงเส้นเดียวแล้วจะทำให้ระบบ Grader ไม่สมบูรณ์ (อาจไม่มีเส้นใดเลยก็ได้) ทั้งนี้คุณไม่ต้องการที่จะหากลอกถามข้อมูลจากผู้คุมระบบบ่อยเกินไป เพราะอาจทำให้ท่านสงสัยได้

ข้อกำหนด

$$1 \leq N \leq 300, N-1 \leq M \leq N(N-1)/2$$

คุณสามารถถามได้ไม่เกิน 15000 คำถาม

การติดต่อ

เราได้ให้ไฟล์ grader.cpp โดยในไฟล์มีฟังก์ชันดังต่อไปนี้เพื่อให้คุณนำไปหาคำตอบที่ต้องการ

```
1. int get_component(std::vector<int> vertices);
```

ฟังก์ชันนี้จะรับอินพุตเป็น vector<int> หมายถึงเซต S ซึ่งเมื่อได้รับอินพุตแล้วฟังก์ชันนี้จะคืนค่าเป็นจำนวนเต็ม C แสดงถึงจำนวนระบบย่อยหลังจากเซิร์ฟเวอร์อื่นๆ ที่ไม่อยู่ใน S ได้ถูกทำลายลงไป

งานของคุณคือต้องเขียนฟังก์ชัน `std::vector<pair<int,int>> find_bridge(int N)` ในไฟล์ `mooz_server.cpp` เพื่อหาเส้นเชื่อมต่อที่ถ้าตัดแล้วจะทำให้ Grader ไม่สมบูรณ์

วิธีการรัน

นำไฟล์ที่ได้มอบให้ไป (grader.cpp, mooz_server.cpp) และ compile ใน terminal/cmd ด้วย คำสั่ง “g++ -std=c++11 grader.cpp mooz_server.cpp -o mooz_server” และ run ด้วย “./mooz_server < input_file_name” โปรแกรมจะ output ความถูกต้องของฟังก์ชันของคุณและจะบอกจำนวนครั้งที่คุณเรียก get_component() ไปด้วย

เกณฑ์การให้คะแนน

ชุดทดสอบจะถูกแบ่งออกเป็นปัญหาย่อย และคะแนนสำหรับปัญหาย่อยใด ๆ จะเป็นคะแนนต่ำที่สุดที่คุณได้รับในชุดทดสอบของปัญหาย่อยนั้น ๆ โดยมีปัญหาย่อยต่าง ๆ ดังต่อไปนี้

ปัญหาย่อย 1 (7 คะแนน): $N \leq 100$ และ $M \leq 500$

ปัญหาย่อย 2 (23 คะแนน): $M \leq 500$

ปัญหาย่อย 3 (25 คะแนน): มีสายเชื่อมต่อระหว่างเซิร์ฟเวอร์ i และ $i+1$ ทุก ๆ $1 \leq i \leq N-1$

ปัญหาย่อย 4 (ไม่เกิน 45 คะแนน): ไม่มีข้อกำหนดเพิ่มเติม

สำหรับปัญหาย่อยที่ 4 จะมีเกณฑ์การให้คะแนนสำหรับแต่ละชุดทดสอบตามจำนวนคำถามที่ถาม หากโปรแกรมของคุณทำงานไม่ทันเวลาหรือให้คำตอบที่ไม่ถูกต้อง จะได้ 0 คะแนน ไม่เช่นนั้น สมมติ โปรแกรมของคุณถาม X คำถาม จะได้คะแนน S_c สำหรับชุดทดสอบดังกล่าวดังต่อไปนี้

- ถ้า $X \leq 6,000$ จะได้ $S_c = 45$ คะแนน
- ไม่เช่นนั้น $S_c = 20 + 22((15000 - x)/9000)^2$

คะแนนสำหรับปัญหาย่อยใด ๆ จะเป็นคะแนนต่ำที่สุดที่คุณได้รับในชุดทดสอบของปัญหาย่อยนั้น ๆ

ไฟล์ข้อมูลนำเข้าสำหรับเกรดเดอร์ตัวอย่าง

ไฟล์ข้อมูลนำเข้าจะขึ้นต้นบรรทัดแรกด้วย N และ M โดย N แทนจำนวนเซิร์ฟเวอร์และ M จำนวนเส้นเชื่อม อีก M บรรทัดระบุข้อมูลเส้นเชื่อมเป็นจำนวนเต็มสามจำนวน X Y B ระบุว่าเส้นเชื่อมระหว่างเซิร์ฟเวอร์ X และ Y และ $B=1$ ถ้าเป็นเส้นเชื่อมที่ถ้าตัดแล้วจะทำให้ Grader ไม่สมบูรณ์ ถ้าไม่เช่นนั้นจะให้ $B=0$

ตัวอย่างไฟล์นำเข้าและ interaction

Input:

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

function calls	return
get_component({1,2,3})	1
get_component({1,2,4})	2
get_component({1,3,4})	1
return {{3,4}}	Correct