

Christmas Tree

[Time limit : 1s] [Memory limit : 128 MB]

วันคริสต์มาสใกล้เข้ามาแล้ว แน่แน่นอนว่าสิ่งที่คุณหาไม่ได้สำหรับวันคริสต์มาสก็คือต้นคริสต์มาสนั่นเอง ในปีนี้คุณก็ไปซื้อต้นไม้มาแล้วต้นหนึ่งเพื่อนำมาทำเป็นต้นคริสต์มาส โดยต้นไม้นี้เป็น unrooted tree ที่มีจำนวน node ทั้งหมด N node และทุก node มีเลขกำกับอยู่ซึ่งมีค่าตั้งแต่ 1 ถึง N ไม่ซ้ำกัน แต่คุณกลับรู้สึกไม่พอใจเพราะมันดูไม่ค่อยสวย คุณจึงอยากที่จะทำให้มันเป็น Binary search tree เพื่อให้ดูสวยงามเข้ากับบรรยากาศ เนื่องจากต้นไม้มีขนาดใหญ่มาก คุณจึงต้องเขียนโปรแกรมเพื่อหาว่าคุณจะสามารถทำต้นไม้ให้เป็น Binary search tree ได้หรือไม่ และถ้าทำได้ ให้ตอบด้วยว่า node ที่เป็น root ของ Binary search tree นั้นคือ node ที่มีเลขใดกำกับ ถ้าเกิดมีหลาย node ที่สามารถเป็น root ได้ ให้ตอบทุก node เรียงจากน้อยไปมาก

Binary search tree มีสมบัติว่า

1. เป็น rooted binary tree (มี root และทุก node มีลูกได้ไม่เกิน 2 ตัว)
2. สำหรับแต่ละ node X
 - 2.1.1 ถ้ามีลูก 2 ตัว จะต้องมิตัวหนึ่งที่เป็นลูกด้านซ้าย และอีกตัวเป็นลูกด้านขวา
 - 2.1.2 ถ้ามีลูก 1 ตัว จะเป็นลูกด้านซ้ายหรือด้านขวาก็ได้
 - 2.2 ถ้ามีลูกด้านซ้าย : ทุก node ใน subtree ด้านซ้ายจะต้องมีค่าน้อยกว่า X
 - 2.3 ถ้ามีลูกด้านขวา : ทุก node ใน subtree ด้านขวาจะต้องมีค่ามากกว่า X

ข้อมูลนำเข้า

บรรทัดแรก ประกอบด้วยจำนวนเต็มบวก 1 จำนวนคือ N แสดงถึงจำนวน node ของต้นไม้ ($2 \leq N \leq 100000$)

จากนั้นอีก $N-1$ บรรทัด แต่ละบรรทัดประกอบด้วยจำนวนเต็ม 2 จำนวนคือ a, b แสดงว่าระหว่าง node a กับ b มี edge เชื่อมกันอยู่ ($1 \leq a, b \leq 100000$ และ $a \neq b$)

รับประกันว่าข้อมูลนำเข้าจะเป็นต้นไม้ที่ถูกต้อง (ไม่มี circuit และทุก node จะมี unique path ไปยัง node อื่น ๆ ทุก node)



ข้อมูลส่งออก

ถ้าต้นไม้สามารถทำเป็น binary search tree ได้ ให้แสดง node ทั้งหมดที่สามารถทำเป็น root ได้ เรียงจากน้อยไปมาก

ถ้าต้นไม้ไม่สามารถทำเป็น binary search tree ได้ ให้แสดงคำว่า IMPOSSIBLE

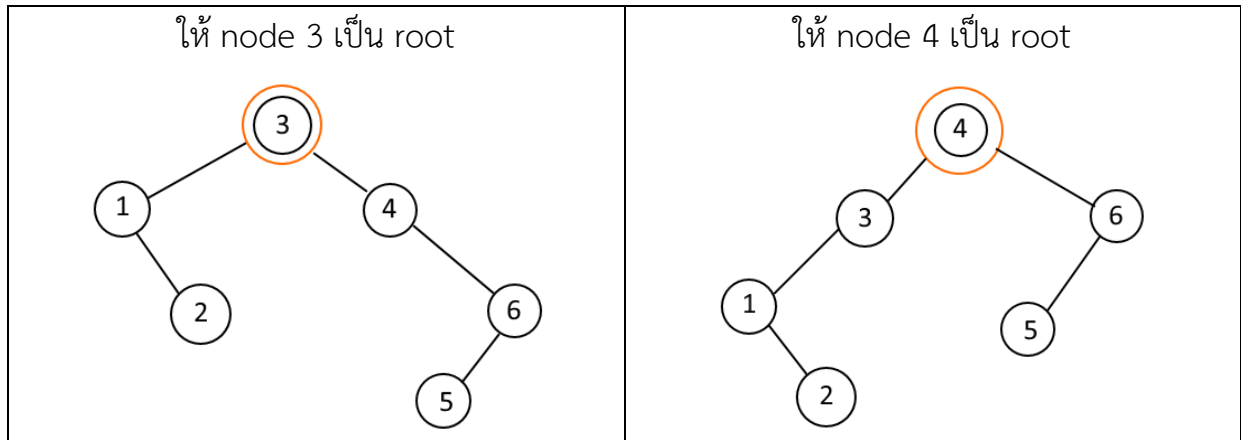
ตัวอย่าง

Input	Output
3 1 2 2 3	1 2 3
4 1 2 2 3 1 4	4
8 1 2 2 3 2 4 4 7 7 5 5 6 7 8	4
6 2 1 1 3 3 4 4 6 6 5	3 4
5 1 2 1 3 1 4 1 5	IMPOSSIBLE
5 3 2 3 4 2 5 4 1	IMPOSSIBLE



คำอธิบายตัวอย่าง

ตัวอย่างที่ 4 เราสามารถสร้าง Binary search tree ได้ 2 แบบ คือ



หมายเหตุ

สำหรับโจทย์ข้อนี้จะมีเทสเคสทั้งหมด 50 ไฟล์ (ปกติ ACM Contest Mode จะมีเพียงไฟล์เดียวแล้วมีเทสเคสย่อยแทน) และผลตรวจที่จะได้รับในการแข่งขัน (ซึ่งจัดเป็นแบบ ACM Contest Mode) นั้น จะทำการเช็คไปเป็นลำดับดังนี้

1. ถ้าโค้ดไม่สามารถคอมไพล์ได้จะได้ผลตรวจเป็น Compilation error
2. ถ้าผลตรวจมีอย่างน้อย 1 เคสที่ได้ X จะได้ผลตรวจเป็น No - Runtime error
3. ถ้าผลตรวจมีอย่างน้อย 1 เคสที่ได้ T จะได้ผลตรวจเป็น No - Time limit exceeded
4. ถ้าผลตรวจมีอย่างน้อย 1 เคสที่ได้ - จะได้ผลตรวจเป็น No - Wrong answer
5. ถ้าผลตรวจเป็น P ทุกเคส ก็จะได้ผลตรวจเป็น Yes

