

ต้นไม้เหมือน

ในข้อนี้ให้นิสิตเขียนคำสั่ง `bool isSameBinaryTree(Tree& t)` เพื่อตรวจสอบว่าต้นไม้ทวิภาคที่ส่งเข้ามา "เหมือน" กับต้นไม้ที่ถูกเรียกคำสั่งนี้หรือไม่ โดยต้นไม้ทวิภาคสองต้นจะ "เหมือน" กันก็ต่อเมื่อ

- ทั้งสองต้นเป็นต้นไม้ว่า หรือ
- ข้อความดังต่อไปนี้เป็นจริง (ให้ A และ B เป็นรากของต้นไม้แต่ละต้น)
 - a. A เก็บข้อมูลเหมือนกับ B และ
 - b. ต้นไม้ย่อยทางซ้ายของ A "เหมือน" กับต้นไม้ย่อยทางซ้ายของ B และ
 - c. ต้นไม้ย่อยทางขวาของ A "เหมือน" กับต้นไม้ย่อยทางขวาของ B

ข้อบังคับ

ห้ามนิสิตใช้บริการใดๆ ของ STL ในส่วนของฟังก์ชัน `bool isSameBinaryTree(Tree& t)`

โจทย์ข้อนี้จะมีไฟล์โปรเจกต์ของ `code::block` ให้ ซึ่งในโปรเจกต์ดังกล่าวจะมีไฟล์ `tree.h` และ `main.cpp` อยู่ ให้นิสิตเขียน code เพิ่มเติมลงไปไฟล์ `tree.h` ในฟังก์ชัน `isSameBinaryTree` และสามารถเพิ่มเติมฟังก์ชันภายในคลาส `Tree` ได้ตามต้องการแต่ห้ามแก้ไขฟังก์ชันหรือตัวแปรใดๆ ที่มีอยู่เดิมโดยเด็ดขาด

ในการส่งไฟล์ขึ้น grader นั้นให้ส่งเฉพาะไฟล์ `tree.h` โดยใน grader จะมี `main` ต่างหากที่ใช้ในการตรวจซึ่งจะทดสอบความถูกต้องของ `isSameBinaryTree` ในกรณีการใช้งานที่ซับซ้อนกว่านี้

คำอธิบายฟังก์ชัน `main()`

โปรแกรมจะรับจำนวนข้อมูล $n \leq 10000$ ในต้นไม้แรกจากคีย์บอร์ด และ รับข้อมูล n ตัวเพื่อเป็นลำดับในการสร้างต้นไม้ต้นแรก จากนั้นจึงรับจำนวนข้อมูล $m \leq 10000$ ในต้นไม้ต้นอีกต้นจากคีย์บอร์ด และ รับข้อมูล m ตัวเพื่อเป็นลำดับในการสร้างต้นไม้ต้น จากนั้นจึงเรียกคำสั่ง `isSameBinaryTree`

ตัวอย่าง

ข้อมูลที่พิมพ์เข้าทาง keyboard	ข้อมูลที่เป็นผลจากการทำงานของโปรแกรม
5 3 4 1 8 2 5 3 1 4 8 2	True
5 3 4 1 2 8 5 3 4 2 1 8	False

คำแนะนำ: Tree จะต้องทำงานถูกต้องไม่ว่า ต้นไม้สองต้นนั้นจะโดนสร้างมาอย่างไรก็ตาม