

Furthest Distance

(1 sec, 512mb)

จงเพิ่มบริการให้กับ CP::map_bst โดยให้เขียนฟังก์ชัน `int furthest_distance()` ซึ่งฟังก์ชันนี้จะต้องคืนค่า “ระยะห่างมากที่สุด” ระหว่าง 2 ปมใด ๆ ในต้นไม้ BST นี้ โดยกำหนดให้ระยะห่างของปม a กับปม b คือ จำนวนเส้นเชื่อมที่เราจะต้องเดินผ่านจากปม a ไปยังปม b

สำหรับต้นไม้ว่างนั้นให้คืนค่า `furthest_distance` เป็น -1

สำหรับต้นไม้ที่มีปมเดียวนั้น ให้คืนค่า `furthest_distance` เป็น 0

ข้อบังคับ

- โจทย์ข้อนี้จะมีไฟล์ตั้งต้นมาให้ ประกอบด้วยไฟล์ `map_bst.h`, `main.cpp` และ `student.h` ให้นักเขียน code เพิ่มเติมลงในไฟล์ `student.h` เท่านั้น และการส่งไฟล์เข้าสู่ระบบ grader ให้ส่งเฉพาะไฟล์ `student.h` เท่านั้น
 - ไฟล์ `student.h` จะต้องไม่ทำการอ่านเขียนข้อมูลใด ๆ ไปยังหน้าจอหรือคีย์บอร์ดหรือไฟล์ใด ๆ
- หากใช้ VS Code ให้ทำการ compile ที่ไฟล์ `main.cpp`

**** main ที่ใช้จริงใน grader นั้นจะแตกต่างจาก main ที่ได้รับในไฟล์ตั้งต้นแต่จะทำการทดสอบในลักษณะเดียวกัน ****

คำอธิบายฟังก์ชัน main

- บรรทัดแรกประกอบด้วยจำนวนเต็ม 1 ตัวซึ่งระบุจำนวนข้อมูลที่จะใส่เข้าไปในต้นไม้
- บรรทัดที่สองประกอบด้วยจำนวนเต็ม n ตัวที่จะใส่เข้าไปในต้นไม้ตามลำดับ
- หลังจากนั้น main จะเรียกฟังก์ชัน `furthest_distance` และแสดงค่าที่ได้คืนมาออกทางหน้าจอ

คำแนะนำ

ข้อนี้สามารถทำได้โดยง่ายโดยเขียนโปรแกรมแบบ recursive และเพื่อให้การเขียนโปรแกรมแบบ recursive ทำได้สะดวก นิสิตสามารถเขียนฟังก์ชัน `int my_recur(node* n, int &aux)` เพื่อกระทำการตามที่นิสิตต้องการได้ ฟังก์ชันดังกล่าวนี้มีโครงอยู่ใน `student.h` แล้ว ถ้าหากนิสิตต้องการจะใช้ สามารถเขียนรายละเอียดของฟังก์ชันดังกล่าวได้เลย

ชุดข้อมูลทดสอบ

- 10% $n \leq 3$
- 15% เฉพาะปมรากเท่านั้นที่อาจจะมีลูกได้มากกว่า 1 ลูก
- 20% ในต้นไม้มีได้อย่างมาก 1 ปมที่มีลูกได้มากกว่า 1 ลูก
- 55% ไม่มีเงื่อนไขอื่นใด

(ตัวอย่างอยู่ในหน้าถัดไป)

ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
2 -6 10	1
10 -1 7 -10 -2 -8 4 1 0 -4 -3	9
10 4 2 8 -3 -5 -7 5 1 -2 -10	7
1 5	0
0	-1