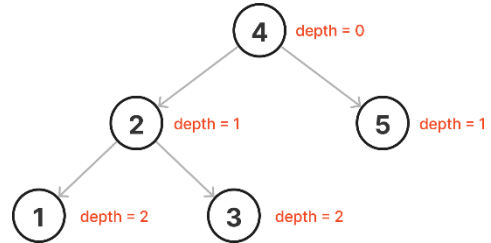


Tree Leaves Depth

(1 sec, 512mb)

จงเพิ่มบริการ long long CP::map_bst::sum_leaves_depth() ซึ่งนับผลรวมของความลึกของใบทั้งหมดในต้นไม้ Binary Search Tree ของเรา โดยกำหนดให้ความลึกของปมใด ๆ ในต้นไม้คือความยาวของ Path จากปมรากไปยังปมดังกล่าว

ตัวอย่างเช่น ต้นไม้ในรูปด้านขวานี้มีความลึกของปมต่าง ๆ เป็นตัวเลขที่กำกับอยู่ด้านข้าง ๆ ปม โดยฟังก์ชันนี้จะต้องคืนผลรวมของความลึกของปมที่เป็นใบทั้งหมดซึ่งมีค่าเป็น 5 (ได้มาจากการนำความลึกของปมหมายเลข 1, 3 และ 5 มารวมกัน)



ข้อบังคับ

- โจทย์ข้อนี้จะมีไฟล์ตั้งต้นมาให้ ประกอบด้วยไฟล์ map_bst.h, main.cpp และ student.h อยู่ ให้นิสิตเขียน 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 ****

คำแนะนำ

ข้อนี้สามารถทำได้โดยง่ายโดยเขียนโปรแกรมแบบ recursive และเพื่อให้การเขียนโปรแกรมแบบ recursive ทำได้สะดวก นิสิตสามารถเขียนฟังก์ชัน long long my_recur(node* n, size_t aux); เพื่อกระทำการตามที่นิสิตต้องการได้ ฟังก์ชันดังกล่าวนี้มีโครงอยู่ใน student.h แล้ว ถ้าหากนิสิตต้องการจะใช้ สามารถเขียนรายละเอียดของฟังก์ชันดังกล่าวได้เลย

คำอธิบายฟังก์ชัน main

main() จะอ่านข้อมูลมา 2 บรรทัด คือ

- บรรทัดแรกประกอบด้วยจำนวนเต็ม N ซึ่งระบุจำนวนข้อมูลใน Binary Search Tree
- บรรทัดที่สองประกอบด้วยจำนวนเต็ม N ตัวคือข้อมูลที่จะใส่เข้าไปใน CP::map_bst ตามลำดับ

ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
5 4 2 5 1 3	5

ชุดข้อมูลทดสอบ

ให้ n คือจำนวนปมในต้นไม้ที่จะเรียกใช้ sum_leaves_depth

- 15% $n \leq 3$
- 15% $n \leq 7$
- 20% $n \leq 50,000$
- 50% $n \leq 500,000$