

BST at level

จงเพิ่มบริการให้กับคลาส CP::map_bst โดยให้เพิ่มฟังก์ชัน std::vector<KeyT> at_level(size_t level) ซึ่งจะทำการคืนค่ารายการ key ทั้งหมดที่อยู่ในที่ ปม ที่อยู่ลึกเป็นระดับชั้นที่ level (โดยให้ปมรากเป็นปมระดับชั้นที่ 0 และปมลูก ๆ ของปมรากเป็นปมระดับชั้นที่ 1)

ค่าที่คืนมานั้นจะต้องเรียงจาก “มากไปน้อย” โดยพิจารณาค่ามากน้อยตาม comparator ของ map_bst ที่เรียกฟังก์ชันนี้

ข้อบังคับ

โจทย์ข้อนี้จะมีไฟล์โปรเจคของ Code::Blocks ให้ ซึ่งในโปรเจคดังกล่าวจะมีไฟล์ map_bst.h, main.cpp และ student.h อยู่ ให้นิสิตเขียน code เพิ่มเติมลงไปไฟล์ student.h เท่านั้น และการส่งไฟล์ขึ้น grader ให้ส่งเฉพาะไฟล์ student.h นิสิตสามารถแก้ไข student.h ได้ โดยอิสระ และสามารถเรียกใช้ฟังก์ชันใด ๆ ใน stl รวมถึงของ map_bst ได้

*** ห้ามทำการพิมพ์ข้อมูลทางจอภาพหรืออ่านข้อมูลจากคีย์บอร์ดในไฟล์ student.h ที่ส่งมายัง grader โดยเด็ดขาด ***

คำแนะนำ

ข้อนี้สามารถทำได้โดยง่ายโดยเขียนโปรแกรมแบบ Recursive และเพื่อให้การเขียนโปรแกรมแบบ recursive ทำได้สะดวก นิสิตสามารถเขียนฟังก์ชัน void my_recur(node* n, size_t level, size_t tmp, std::vector<KeyT> &v); เพื่อกระทำการตามที่นิสิตต้องการได้ ฟังก์ชันดังกล่าวนี้มีโครงอยู่ใน student.h แล้ว ถ้าหากนิสิตต้องการจะใช้ สามารถเขียนรายละเอียดของฟังก์ชันดังกล่าวได้เลย

คำอธิบายฟังก์ชัน main()

โปรแกรมจะเริ่มต้นจาก CP::map_bst<int,bool> m ซึ่งเป็น bst ว่าง หลังจากนั้น main จะทำการอ่านข้อมูลจากคีย์บอร์ดดังรูปแบบต่อไปนี้

บรรทัดแรกประกอบด้วยจำนวนเต็มตัวแรกคือ n ซึ่งระบุจำนวนข้อมูลที่ต้องการใส่เข้าไปใน bst

บรรทัดที่ 2 ประกอบด้วยจำนวนเต็ม n ค่า คือข้อมูลที่จะอยู่ใน bst โดย main จะเพิ่มข้อมูลดังกล่าวเข้าไปใน bst ตามลำดับ

บรรทัดที่ 3 ประกอบด้วยค่า k

บรรทัดที่ 4 ประกอบด้วยจำนวนเต็ม k ตัว โดยที่แต่ละตัวคือค่า level ที่จะเรียกใช้ m.at_level(level) และทำการพิมพ์ข้อมูลที่ฟังก์ชันดังกล่าวคืนค่าออกมาตามลำดับ

*** main ใน grader นั้นจะแตกต่างจาก main ที่นิสิตได้รับ แต่จะเป็นการทดสอบในลักษณะเดียวกัน ขอให้เขียนฟังก์ชันเพิ่มเติมให้ตรงตามนิยามที่กำหนดไว้ข้างต้น ***

ชุดข้อมูลทดสอบ

- 10% mSize = 3, level = 1 เสมอ
- 10% mSize = 4, level = 1 หรือ 2
- 20% level <= 3
- 20% ไม่มีข้อจำกัดใด ๆ
- 40% ส่วนของโปรแกรมของนิสิตจะต้องไม่มีการเรียกใช้ mLess โดยเด็ดขาด (ซึ่งรวมถึงการนำ mLess ไปใช้ในการเรียงข้อมูล หรือสร้าง object อื่น ๆ)

ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
5	level 0 size = 1: 1
1 2 3 -2 -3	level 1 size = 2: 2 -2
3	level 2 size = 2: 3 -3
0 1 2	