

## Subtree Extract

จงเขียนฟังก์ชันเพิ่มเติมให้กับคลาส `CP::map_bst` คือฟังก์ชัน `pair<KeyT,MappedT> subtree(map_bst<KeyT,MappedType> &left, map_bst<KeyT,MappedType> &right)` โดยเมื่อต้นไม้ BST ที่เรียกฟังก์ชันนี้เป็นต้นไม้ว่าง ฟังก์ชันนี้ต้องไม่ทำงานใด ๆ และคืนค่า `pair<KeyT,MappedT>()` มา แต่สำหรับในกรณีที่ต้นไม้ไม่ว่างนั้นฟังก์ชันนี้จะดำเนินการดังต่อไปนี้

- ย้าย left subtree ของรากไปเก็บไว้ใน left (ข้อมูลอื่นใดใน left ที่มีอยู่ก่อนหน้าต้องโดนลบทิ้ง) พร้อมทั้งตั้งค่า `mSize` ของ left ให้ถูกต้อง
  - ย้าย right subtree ของรากไปเก็บไว้ใน right (ข้อมูลอื่นใดใน right ที่มีอยู่ก่อนหน้าจะต้องโดนลบทิ้ง) พร้อมทั้งตั้งค่า `mSize` ของ right ให้ถูกต้อง
  - คืนค่าที่เก็บไว้ในปมราก
  - หลังจากเรียกฟังก์ชันนี้ ต้นไม้ที่เรียกจะเหลือปมเพียงปมรากอยู่ในต้นไม้เท่านั้น
  - การย้าย left subtree และ right subtree นั้นจะต้องไม่เปลี่ยนแปลงโครงสร้างของ left subtree และ right subtree กล่าวคือ หลังจากเรียกแล้ว left จะมีโครงสร้าง และ ข้อมูล หน้าตาเหมือนกับ left subtree ก่อนเรียก และ right จะมีโครงสร้าง และ ข้อมูล หน้าตาเหมือนกับ right subtree ก่อนเรียก
- การที่จะได้คะแนนเต็มในข้อนี้ ฟังก์ชันนี้ควรจะใช้เวลาเป็น  $O(\text{left.size()} + \text{right.size}())$

## ข้อควรระวัง

การแก้ไข node ของต้นไม้จะต้องคำนึงถึงตัวแปร left, right parent ให้ถูกต้องด้วย

## ข้อบังคับ

- ในโจทย์ข้อนี้ มีโครงฟังก์ชัน `size_t process(node *n)` เพื่อไว้ให้ใช้สำหรับกรณีที่ต้องการเขียนฟังก์ชันใด ๆ ที่ทำงานกับ node ต่าง ๆ นิสิตสามารถใช้ฟังก์ชันนี้สำหรับทำอะไรก็ได้
- โจทย์ข้อนี้จะมีไฟล์โปรเจ็คของ `Code::Blocks` ให้ ซึ่งในไฟล์โปรเจ็คดังกล่าวจะมีไฟล์ `map_bst.h`, `main.cpp` และ `student.h` อยู่ให้นิสิตเขียน code เพิ่มเติมลงในไฟล์ `student.h` เท่านั้น และการส่งไฟล์เข้าสู่ระบบ grader ให้ส่งเฉพาะไฟล์ `student.h` เท่านั้น
  - ในไฟล์ `student.h` ดังกล่าวจะต้องไม่ทำการอ่านเขียนข้อมูลใด ๆ ไปยังหน้าจอหรือ คีย์บอร์ดหรือไฟล์ใด ๆ

## คำอธิบายฟังก์ชัน main()

main จะอ่านข้อมูลมาสองบรรทัด ตามรูปแบบนี้

- บรรทัดแรกประกอบด้วยจำนวนเต็ม n
- บรรทัดที่สองประกอบด้วยจำนวนเต็มจำนวน n ตัว คือ กำหนดให้ข้อมูลเหล่านั้นคือ a1 ถึง an หลังจากนั้น main จะสร้าง `map_bst<int,int> m` ขึ้นมา และนำค่า i ไปใส่ไว้ใน `m[ai]` หลังจากนั้น main จะเรียก `m.split(left,right)` แล้วทำการพิมพ์ค่าของ left, right และ ค่าที่คืนมาจาก subtree ออกทางหน้าจอ

## ชุดข้อมูลทดสอบ

- 20% ต้นไม้จะมีขนาดไม่เกิน 3
- 30% ต้นไม้จะมีจำนวนข้อมูลใน left subtree เท่ากับจำนวนข้อมูลใน right subtree
- 50% ไม่มีข้อกำหนดอื่นใด