

Generate Optimal Binary Search Tree

เราทราบว่าโครงสร้างของ Binary Search Tree ที่มีข้อมูลเหมือนกันนั้น มีได้หลายแบบ ในโจทย์ข้อนี้ เราต้องการจะสร้าง Binary Search Tree ที่มีข้อมูลส่วน key เป็นค่าตั้งแต่ 1 ถึง $n = 2^k - 1$ ที่มีโครงสร้างที่ดีที่สุด กล่าวคือ ต้นไม้ bst ต้องมีความสูงเป็น $k-1$ จงเขียนฟังก์ชัน `void gen_best_bst(int n, CP::map<int,int> &bst)` ที่ทำการใส่ข้อมูลที่มีค่า key เป็น 1 ถึง $n = 2^k - 1$ ลงไปใน bst ในลำดับที่ทำให้ bst มีความสูงเป็น $k-1$ (ส่วนค่า value นั้นเป็นอะไรก็ได้)

ข้อบังคับ

โจทย์ข้อนี้จะมีไฟล์ `best_bst.cpp` มาให้แล้ว โดยในไฟล์ดังกล่าวจะมีโครงของฟังก์ชัน `gen_best_bst` ซึ่งเป็นฟังก์ชันที่จะต้องเขียน และมีฟังก์ชัน `main` พร้อมด้วยคลาส `CP::map` อยู่แล้ว ห้ามทำการแก้ไขใด ๆ กับฟังก์ชัน `main` หรือคลาส `CP::map` โดยเด็ดขาด แต่สามารถเขียนฟังก์ชันอื่น ๆ เพิ่มเติมนอกคลาส `CP::map` ได้

คำอธิบายฟังก์ชัน `main()`

`main` จะทำการอ่านข้อมูลค่า “ความสูง” (ความสูงคือ $k-1$) จาก keyboard แล้วคำนวณค่า n และเรียกฟังก์ชัน `gen_best_bst` เพื่อสร้าง bst หลังจากนั้น `main` จะทำการแสดงข้อมูล key ของต้นไม้ bst ดังกล่าวในรูปแบบที่แปรตามโครงสร้างของต้นไม้ กล่าวคือ `main` จะทำการท่องต้นไม้วิธีการแบบ preorder, inorder และ postorder และพิมพ์ค่าของ key ในปมที่วิ่งผ่านตามการท่องต้นไม้แบบดังกล่าวตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
2 7	4 2 1 3 6 5 7 1 2 3 4 5 6 7 1 3 2 5 7 6 4