

การจำลองการทำงานของโครงสร้างข้อมูล Binary Search Tree

กำหนดคลาส BSTNode ในไฟล์ BSTNode.cpp ทำหน้าที่เก็บข้อมูลค่าที่อยู่ในโหนดของ Binary Search Tree และมี Pointer ชี้ไปยังลูกคนซ้าย (BSTNode *left) และ ลูกคนขวา (BSTNode *right)

หน้าที่ของเราคือเขียนฟังก์ชันในคลาส BST ซึ่งเก็บข้อมูลเป็น Pointer ไปยังโหนดแรกของ Binary Search Tree (BSTNode *root) โดยแต่ละฟังก์ชันมีรายละเอียดดังต่อไปนี้

- **void insert(int value)** เพิ่มโหนดใหม่ที่มีค่า value ให้กับ Binary Search Tree โดยค่าที่เพิ่มจะไม่ซ้ำกับค่าที่มีอยู่ใน Binary Search Tree อยู่แล้ว
- **void remove(int value)** ลบโหนดที่มีค่า value ออกจาก Binary Search Tree
- **int get_depth(int value)** คืนค่า depth ของโหนดที่มีค่าเท่ากับ value หากไม่มีค่า value อยู่ใน Binary Search Tree ให้คืนค่า -1

หมายเหตุ: นักศึกษาสามารถดูรายละเอียดการทำงานของแต่ละฟังก์ชันในสไลด์ 251Ch10tree_1.pdf และ 251Ch10tree_2.pdf

ขั้นตอนการทำงานบ้าน

1. ดาวน์โหลดไฟล์การบ้านได้ที่ <https://github.com/CS-CMU/cs252student>
2. ในโฟลเดอร์ HW08 จะมีไฟล์ main.cpp, BSTNode.cpp และ HW08.cpp
3. ไฟล์ main.cpp จะเป็นไฟล์สำหรับทดสอบโปรแกรมของนักศึกษา สามารถแก้ไขได้ตามความเหมาะสม ไฟล์นี้ไม่ต้องส่งและไม่มีการตรวจ
4. ไฟล์ BSTNode.cpp มีนิยามของคลาส BSTNode โดยมี
 - ตัวแปร int value ที่เก็บค่าของโหนด
 - ตัวแปร BSTNode *left ที่ชี้ไปยังโหนดลูกคนซ้าย
 - ตัวแปร BSTNode *right ที่ชี้ไปยังโหนดลูกคนขวา **ไฟล์นี้ห้ามแก้ไข แม้จะไม่ต้องส่งก็ตาม**
5. ไฟล์ HW08.cpp จะเป็นไฟล์ที่นักศึกษาต้องแก้ไข โดยมีรายละเอียดดังนี้

คลาส BST ประกอบไปด้วย

1. ตัวแปร Pointer `BSTNode * root` ต้องประกาศเป็น `public` เท่านั้น (ใช้สำหรับการตรวจ)
2. Constructor สำหรับคลาส BST (เขียนให้แล้ว)
3. ฟังก์ชัน `void insert(int value)` เพิ่มโหนดใน BST
4. ฟังก์ชัน `void remove(int value)` ลบโหนดจาก BST
5. ฟังก์ชัน `void get_depth(int value)` คืนค่า `depth` ของโหนด

ฟังก์ชันที่นักศึกษาต้องเขียนส่งคือ `insert()`, `remove()` และ `get_depth()`

นักศึกษาสามารถสร้างตัวแปรหรือฟังก์ชันเพิ่มได้ตามความเหมาะสม (ไม่ตรวจ)

6. ส่งไฟล์ `HW08.cpp` ที่ <https://grader.cs.science.cmu.ac.th> คะแนนที่ได้ในเว็บเกรดเดอร์คือคะแนนที่นำไปใช้ตัดเกรด (หลังจากรันโปรแกรมจบการลอกแล้ว)

ตัวอย่างการรันโปรแกรมที่ถูกต้อง



```
main.cpp x
1 #include<iostream>
2 #include "HW08.cpp"
3 using namespace std;
4
5 int main() {
6     BST tree;
7
8     tree.insert(2);
9     tree.insert(1);
10    tree.insert(3);
11    tree.remove(2);
12    cout << tree.get_depth(1) << endl;
13    cout << tree.get_depth(2) << endl;
14    cout << tree.get_depth(3) << endl;
15
16    return 0;
17 }
```

```
Console Shell
> clang++-7 -pthread -std=c++17 -o main BSTNode.cpp H Q X
p main.cpp
> ./main
1
-1
0
> []
```