

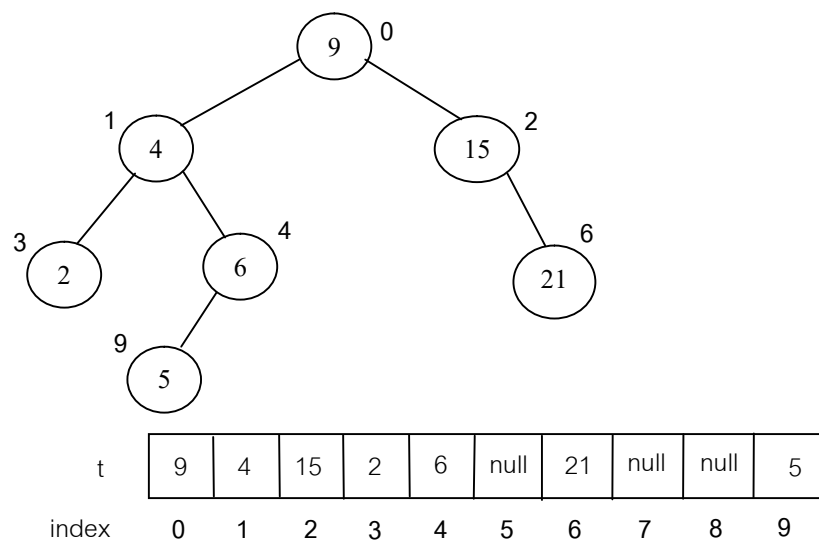
CN 202 – Data Structures and Algorithms I

Solution for Homework 13

1. (ข้อสอบปลายภาคปี 55)

สิ่งที่กำหนดให้

- (a) Java Interface ที่มีชื่อว่า Treeltem.java และ BSTADT.java ตามลำดับ โดยทั้งสอง Interface เป็นไปตามที่ได้เรียนในชั้นเรียน
- (b) Java Class ที่มีชื่อว่า BST.java ซึ่ง Implement โครงสร้างของ Binary Search Tree ด้วย Array ของ Treeltem (กล่าวคือ Physical Model ของ Binary Search Tree คือ Array ของ Treeltem) ดังแสดงด้วยตัวอย่างต่อไปนี้



- แต่ละ Node ของ Binary Search Tree จะสอดคล้องกับแต่ละ Index ของ Array โดย
 - Node ที่ Depth = 0 (Root) สอดคล้องกับ Index 0
 - Node ที่ Depth = 1 สอดคล้องกับ Index 1 และ 2
 - Node ที่ Depth = 2 สอดคล้องกับ Index 3, 4, 5, และ 6ฯลฯ
- ถ้าโครงสร้างของ Binary Search Tree ไม่มี Node ที่ตำแหน่งใด ค่าของ Array ที่มี Index ที่สอดคล้องกับ Node นั้น จะมีค่าเป็น null

```

public class BST implements BSTADT
{
    private Treeltem[] t;

    //constructor

    public BST(int maxSize)

    {
        t = new Treeltem[maxSize];

        t[0] = null; // Empty Tree

    }

    // methods

    private int getLeftChild(int i) { }

    private int getRightChild(int i) { }

    ...

    public Treeltem find(Object k) { }

}

```

สิ่งที่ต้องการให้แก้ปัญา

- (ก) ให้เขียนรายละเอียดการทำงานของ Private Method ที่ชื่อ getLeftChild โดย Method นี้จะรับค่า i ซึ่งเป็น Index ของ Array t ที่สอดคล้องกับ Node หนึ่ง ๆ ของ Binary Search Tree จากนั้น Method นี้ จะทำการหาค่า Index ที่สอดคล้องกับ Left Child ของ Node นี้ และส่งค่า Index ที่หาได้นี้ ออกมาเป็น Output ของ Method

ตัวอย่าง จากรูปตัวอย่างโครงสร้างของ Binary Search Tree และ Array ที่กำหนดให้

เมื่อทำการเรียก getLeftChild(0) จะได้ 1 เป็น Output

getLeftChild(4) จะได้ 9 เป็น Output

- (ข) ให้เขียนรายละเอียดการทำงานของ Private Method ที่ชื่อ getRightChild โดย Method นี้จะรับค่า i ซึ่งเป็น Index ของ Array t ที่สอดคล้องกับ Node หนึ่ง ๆ ของ Binary Search Tree จากนั้น Method นี้จะทำการหาค่า Index ที่สอดคล้องกับ Right Child ของ Node นี้ และส่งค่า Index ที่หาได้นี้ ออกมาเป็น Output ของ Method

ตัวอย่าง จากรูปตัวอย่างโครงสร้างของ Binary Search Tree และ Array ที่กำหนดให้

เมื่อทำการเรียก getRightChild(0) จะได้ 2 เป็น Output

getRightChild(1) จะได้ 4 เป็น Output

- (ค) ให้ปรับเปลี่ยนรายละเอียดการทำงานของ Method ที่ชื่อ find จากที่ได้เรียนในชั้นเรียน เพื่อให้การทำงานของ find สามารถทำงานได้ผลอย่างถูกต้องกับโครงสร้าง Binary Search Tree ที่ถูกสร้างด้วย Array ของ Treeltem โดยเขียนรายละเอียดการทำงานทั้งหมดของ find มาเป็นคำตอบ

ข้อกำหนดในการแก้ปัญาข้อ (ค)

- Method find ต้องมีการเรียกใช้ Method getLeftChild ในข้อ (ก) และ getRightChild ในข้อ (ข) ช่วยในการทำงาน

ข้อควรระวังในการแก้ปัญหาข้อ (ค)

- ในระหว่างการค้นหาค่า Key k ภายใน Array t หากพบว่า ค่า Index ของ Array ที่สอดคล้องกับ Left Child หรือ Right Child มีค่ามากกว่า Index สูงสุดของ Array t (กล่าวคือ มากกว่า $t.length - 1$) ให้สรุปได้ว่า ไม่มี Key k อยู่ใน Binary Search Tree ดังนั้น ในกรณีนี้ Method find จะให้ Output เป็น null

```
// BST (Implemented with Array of Treeltems)
private int getLeftChild(int i)
{ return (2 * i) + 1; }

private int getRightChild(int i)
{ return (2 * i) + 2; }

// find and return the item with desired key k
// return null if not found
public Treeltem find(Object k)
{
    if (t[0] == null)
        return null; // null is returned if tree is empty

    int current = 0;
    Treeltem currentItem = t[current];
    while (currentItem.compareTo(k) != 0)
    {
        if (currentItem.compareTo(k) > 0)
            // k < key(currentItem)
            current = getLeftChild(current);
        else
            // k >= key(currentItem)
            current = getRightChild(current);
        if ((current > (t.length-1)) || (t[current] == null))
            return null; // can't find it
        else
            currentItem = t[current];
    }
    // the item with desired key is found
    return currentItem;
}
```