Computer Programming Assignment 3

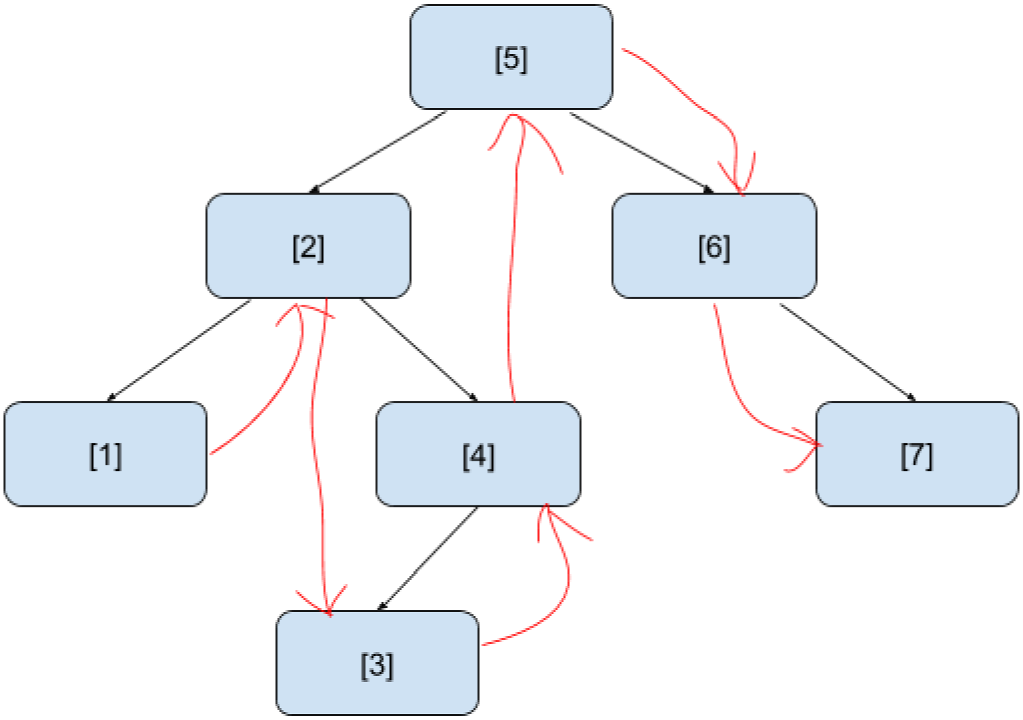
Checkpoints

1. You should do the assignment in your own. You are not allowed to share code with others and/or copy code from other resources. If you are caught, as in the syllabus, you will get a failing grade.
2. Grading will be done in the Linux environment using Java 10.
3. Program failed to compile/run will result 0.
4. Do not loop your program to repeat unless you are told so.
5. Do not change input/output format unless you are told so.
6. Write your name and student number at top of program as a comment.
7. Do not include Korean (and any other language than English) comment. In some encoding formats, Korean comments will cause compilation errors in the Linux environment, which will result in a 0 for your grade.
8. If you have any questions, please contact cp2018ta@tcs.snu.ac.kr

Submission

1. Submit your assignment on eTL.
2. Zip your file (or tar) as '<Student ID>-assign3.zip'
    a. ex.) 2017-12345-assign3.zip
3. Due date of this assignment is Nov 16th, 2018, 23:59:59
4. No late submission is allowed.

Problem 1   Linked List embedded Binary Search Tree



The “next” fields are shown in red colored arrows.

You will implement a binary tree where each node contains the four fields: val, left and right fields to provide the functionality of a binary search tree, and next field to provide the functionality of a linked list.
(Assume the class name for a node is `LBST_node` and the linked binary search tree is `LinkedBinarySearchTree`.)

```
class LBST_node{
  public int val;
  public LBST_node l_child;  //Node’s class name is BT_node
  public LBST_node r_child;
  public LBST_node next;
   /**constructors and auxiliary methods can be added here**/
}
```

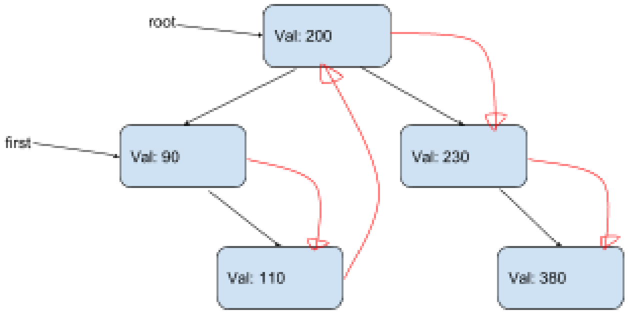
The three fields(val, l_child, r_child) will be used for Binary Search Tree, and the other field(next) will be used to traverse the tree in in-order.

You also need to complete the following five member methods: insert, remove, search, range_search and list.

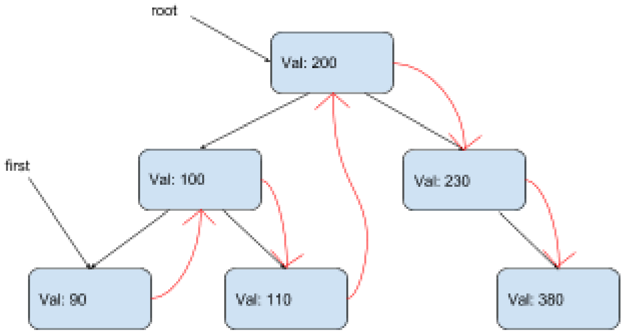
```
class LinkedBinarySearchTree{
  public LBST_node root;  //For easier grading, it is public member.
  public LBST_node first;  //The leftmost node

  public void insert(int num);
  public void remove(int num);
  public boolean search(int num);
  public boolean range_search(int left_val, int right_val, int num);
  public LBST_node[] list(){...};
  /**constructors and auxiliary methods can be added here,
   * not fields.
   **/
}
```

- “root” stores the root of binary tree.
- “first” stores a reference to the node containing the smallest value in the tree -- to be utilized as the head of the linked list.
- “insert(int num)” inserts a number so that binary search tree structure is maintained. When the same number already exists in the tree, nothing will be done.

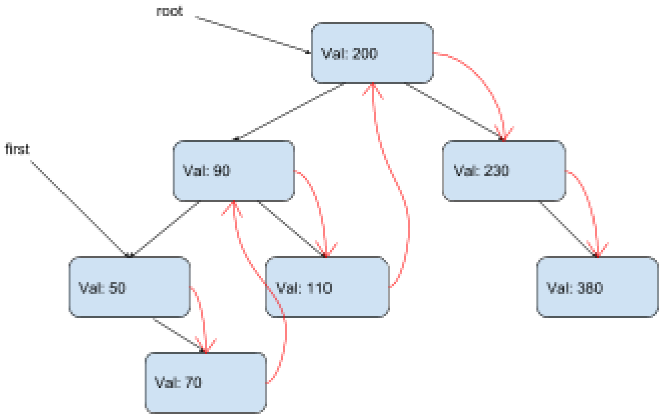


Before Insertion

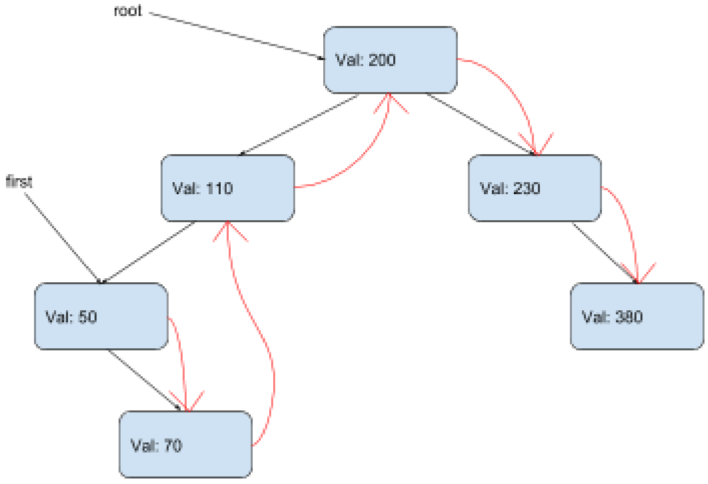


After Insertion of Val: 100

- “remove(int num)” removes an existing number from the tree. When there is no such a number, nothing will be done.



Before Remove of 90



After Remove of 90

- “search(int num)” should search through binary tree to find out whether the number is in the tree or not. If it exists, return true and return false otherwise.

- “range_search(int left_val, int right_val, int num)” should search a number from “left_val(inclusive)” to “right_val(inclusive)”. You can make it easier with the “next” field
Assume the first range index is 0.

e.g.)

```
LinkedBinarySearchTree lbst_tree
boolean test;
test = lbst_tree(2, 4, 10);    //search number 10 in [2,4] range
```

- “list()” should return all the nodes in the tree in sorted order through array “LBST_node[]” by using next fields.

e.g.)

```
LinkedBinarySearchTree lbst_tree;
… //inserting elements (10, 20, 30, 40, 50, 55)
LBST_node node_list[] = lbst_tree.list();
for(int i = 0; i < node_list.length; i++){
  System.out.println(node_list[i]);
}
```

[Output]

```
10
20
30
40
50
55
```

Submission

LBST_Node.java, LinkedBinarySearchTree.java
(or more if you need)

Grade

```
javac *.java
java Assign3_grade1
```

The “Assign3_grade1” class’s main method will test your classes’ methods.