**Question:**

In this question you should complete some methods in **BSTree.java** file.

The class **User** with 3 data members: name, age, and id are given and you do not need to edit it.

The **BSTree** class is a binary search tree of **User** objects.

The variable **id is the key of the tree, thus it must be unique**.

(**This means that “id” is the information to decide whether a node is smaller/larger than the other**)

The following methods should be completed:

* Void **insert(xName, xAge,xId)**- insert new User with name=xName, age=xAge, and color=xColor to the tree (age>0)
* Void **f1()**- You do not need to edit this function. Your task is to complete the insert(…) function above only. Output in the file **f1.txt** must be the following:

Content of the file f1.txt:

(A,50,8) (B,10,2) (E,30,10) (C,4,5) (F,14,9) (H,34,11) (D,25,4) (G,7,6) (J,19,3) (I,20,7)

(B,10,2) (J,19,3) (D,25,4) (C,4,5) (G,7,6) (I,20,7) (A,50,8) (F,14,9) (E,30,10) (H,34,11)

* Void **f2()**- Perform the post-order traversal from the root but display to file f2.txt nodes with age<25 only.

(Hint: Copy the function postOrder(…) to function postOrder2(…) and modify it.)

Output in the file **f2.txt** must be the following:

Content of the file f2.txt:

(A,50,8) (B,10,2) (C,4,5) (D,25,4) (J,19,3) (G,7,6) (I,20,7) (E,30,10) (F,14,9) (H,34,11)

(J,19,3) (I,20,7) (G,7,6) (C,4,5) (B,10,2) (F,14,9)

* Void **f3()** – add age of all nodes have only one child to 3 (age=age+3).

(**Hint**: such nodes will have either left child or right child.)

Output in the file **f3.txt** must be the following:

Content of the file f3.txt:

(A,50,8) (B,10,2) (E,30,10) (C,4,5) (F,14,9) (H,34,11) (D,25,4) (G,7,6) (J,19,3) (I,20,7)

(A,50,8) (B,13,2) (E,30,10) (C,4,5) (F,14,9) (H,34,11) (D,28,4) (G,10,6) (J,19,3) (I,20,7)

* Void **f4()**- Calculate height of last node in postOrder traversal from the root.

(**Hint**: Calculate height of the node (height of a node is height of subtree with root is the node which we need to calculate height, assume that if there is only one node then height = 1).)

Output in the file **f4.txt** must be the following:

Content of the file f4.txt:

(A,50,8) (B,10,2) (C,4,5) (D,25,4) (J,19,3) (G,7,6) (I,20,7) (E,30,10) (F,14,9) (H,34,11)

5

* Void **f5()**- Reset age of all nodes locate in left branch of 1st node in preOrder traversal from root to zero(age=0).

(**Hint**: Traversal all nodes in left branch the node and set age of each node to 0.)

Content of the file f5.txt:

(B,10,2) (J,19,3) (D,25,4) (C,4,5) (G,7,6) (I,20,7) (A,50,8) (F,14,9) (E,30,10) (H,34,11)

(B,0,2) (J,0,3) (D,0,4) (C,0,5) (G,0,6) (I,0,7) (A,50,8) (F,14,9) (E,30,10) (H,34,11)

* Void **f6()**- Find the node which has biggest id (right most) node in the tree. The output of **f6.txt** is as follows:

Content of the file f6.txt:

(B,10,2) (J,19,3) (D,25,4) (C,4,5) (G,7,6) (I,20,7) (A,50,8) (F,14,9) (E,30,10) (H,34,11)

(H,34,11)

**Submission:**

* (1) clean and build project
* (2) rename folder “dist” to “run”
* (3) compress the folder “Given” and rename to “studentID.rar/zip” (ex: SE17123456.rar or SE16654321.zip)
* (4) submit the rar/zip file to lms
* Deadline: 14h00