Computing Systems Lab

IT69101

(Autumn 2010)

Assignment #4 25/08/2010

Instructions:

- For all programs in this assignment, consider the input data files you have followed in Assignment #3.
- For each execution for Problem 1 and 2, you should display the final tree using GraphViz.

(a) ***void insertNodeBST(*root, int data)*:** To insert a node with value *data* into an existing binary search tree with pointer to root node is **root*. The tree may be empty, and data may exist.

(b) ***int searchNodeBST(*root, int data)***: T search for a node with value data in an existing binary search tree with pointer to root node is **root.* The tree may be empty, and data may not exist. The method returns 0 if a node with value *data* does exist, otherwise 1.

(c) ***void mergeBST(*root1, *root2)***: To merge a binary search tree with pointer to root is **root2* to another binary search tree with pointer to root **root1*. Note that either or both may be null tree(s) and some data may common in the two trees. You should use the methods *void insertNodeBST(*root, int data)* and *int searchNodeBST(*root, int data)* in order to implement the method *void mergeBST(*root1, *root2).*

(d) ***int computeHeigh(*node, *root)***: Compute the height of any node with pointer to it is **node* in any binary tree whose pointer to root node is **root*.

Continuing with the problem in Assignment 3, 1(a), have any binary search tree. For such a tree you call the above methods.

Use the method *int computeHeigh(*node, *root)* to calculate the height of any binary tree. Also check whether a given binary search tree is a height balanced or not. You should calculate the balance factor of each node and then store them in each node.

Last date of submission: 01/09/2010