ITCS 2215 Project 3-2: Binary Search Tree:Height, Bal. Factor Spring 2016

Due Friday April 29, 11.59pm

In part 2 of the project, you will use the binary search tree from part 1, compute the height and balance factors at each node of the tree and store it in an object (using the generic parameter and methods provided by BRIDGES). You will display the height and balance factors by loading the label with the information, so that the visualization can display these during mouseover operations on the nodes.

1. [**Computing Tree Height:**] You will compute the height of each tree node and store it at that node. The height of the tree at any node rt is given by a recursive function, for instance,

   ```
   int Height_Of_BST(BSTElement rt) {
   if (rt == NULL)
       return -1
   else
       return 1 + MAX(Height_of_BST(rt.getLeft()), Height_of_BST(rt.getRight());
   }
   ```

   For a single node tree, the height is zero and a null tree will return -1.

   You will need to also store the height and balance factor at each node.

2. [**Balance Factor.**] You will also compute and store the balance factors at each node, defined as

$$Balance\_Factor = Height_{right\_subtree} - Height_{left\_subtree}$$

[Requirements.]

1. Create a simple class to hold the Height and Balance factor and make that the generic object in the BSTElement class. Augment the label with the height and balance factors, so that this can be visualized with the tree, via mouseover operations.

2. You should ensure that you use a single call to your Height computation function to update all nodes with their heights and balance factors. You may add this function to the BST class.

Evaluation.

By interactive Demo.

Grading Rubric.

1. Height, Balance Factor computation(3 points)

2. Visualization of height and balance factors (1 points)

3. Documentation(1 point)