ITCS 2215   Project 3-1: Binary Search Tree with EQuake Data   Spring 2016

Due Monday April 25, 11.59pm

In this project, you will use a provided Binary Search Tree API to build a tree containing earthquake dataset that will be provided as part of the BRIDGES API. In the following parts you will adapt your implementation to maintain a balanced (AVL) tree, performing rotations to restore balance after each specified insert (or delete) operation.

You will be provided with a simple driver that will extract the earthquake data via the BRIDGES server. You will then use the BST API to insert this into a binary search tree. You will need to familiarize yourself with the BSTElement and its ancestor classes (TreeElement, Element), in preparing for this project, as well as the provided BST API.

You will perform the following tasks:

1. [**Before Coding:**] Review the BRIDGES Java API for **BSTElement, TreeElement, Element** and the provided BST API. Also review the **EarthquakeUSGS** class that defines an earthquake record.

2. [**Basic Tasks.**]

    (i) Using the provided driver, specify the number of earthquake records to extract from the server (the getAssociations() method retrieves a specified number of earthquake records). Each earthquake record will become part of the 'data' component of BSTElement(generic parameter of Element).

    (ii) Use the provided BST API to insert each earthquake record into a binary search tree. Use the magnitude of the quake as a search key.

    (iii) Run the driver and visualize the results. You should see a binary tree with search keys beside each node.

3. **Operations on the Tree.** Set up a user input loop to support the following 3 operations:

    (i) **'Largest'.** Write a method to find the quake of the largest magnitude and highlight the node. Visualize the results.

    (ii) **'Smallest'.** Write a method to find the quake of the smallest magnitude and highlight the node. Visualize the results.

    (iii) **'Range min max'.** Specify a magnitude range (min and max are between 0.0 and 10.); write a method to highlight all the nodes that are in this range. Visualize the results.

Evaluation.

By interactive Demo.

Grading Rubric.

1. Basic Tasks (2 points)

2. Largest (2 points)

3. Smallest (2 points)

4. Range (3 points)

5. Documentation(1 point)