# ITCS 2214-002 Program 2.1 – Binary Search Trees Spring 2016

## Due Thursday April 21st, 2016 at 11:55PM

*This assignment is the second and final portion of the Tree project for this course. The primary goal is to use Bridges to read a sequence of Earthquake Tweets from the USGS Earthquake Twitter feed, add some more traversal methods to those already implemented in Program 2.0, and then visualize the results.*

## Overview

The primary goals of this program are as follows:

(1) Add a few more methods to the BST implementation from Program 2.0
(2) Use the skeleton code to read a list of Tweets and store them in a BST

The **Driver** will create an instance of the BST class, read a sequence of Earthquake tweets (with attributes including magnitude, location, date) from Bridges and store them in the Tree with the earthquake magnitude as the search key, and allow user-specified commands to perform various operations on the tree.

Note: this assignment requires you to use a modified JAR file supporting the EarthquakeUSGS Tweet class. A link to the JAR will be provided on this assignment's page on Moodle.

## Tasks

**Copy** your package from program 2.0 and **download** the new program2_1_Skeleton driver from Moodle

**Add** the new Bridges JAR file to the build path for this project. A link to the JAR will be provided on this assignment's page on Moodle.

**Documentation** for Bridges classes can be found at the following link:
http://bridgesuncc.github.io/doc/java-api/current/

**BridgesBST –**

**Modify your implementation from Program 2.0**

When you instantiate your Tree, it will be parameterized to hold <Double, EarthquakeUSGS>; the *key* is the magnitude of the Earthquake, and the *value* is the EarthquakeUSGS object containing all attributes of each quake. These objects will be imported through Bridges. See the skeleton code for further details.

- Add methods to find and highlight the **min** and **max** nodes

- Add a method to **reset** all nodes in the tree to a particular color and/or opacity

- Add a method to **setLabels**, traversing the entire tree and setting each node's label to show location and date of the earthquake. *Note that the value at each* *node contains EarthquakeUSGS objects, and each has getLocation and getDate methods.*

- Add a method to **findLocation**, traversing the entire tree and highlighting all nodes whose locations contain a user-specified string location. *For example, if a user enters 'Alaska,' then all nodes with 'Alaska' in the location string (ignore case) should be highlighted and all other nodes should be ignored. You can use opacity to make the highlighted nodes prominent.*

- Add a method to **highlightRange**. Allow the user to specify the min and max magnitudes then highlight all nodes falling within that range of magnitudes.

**Driver –**

**Download the skeleton driver from Moodle and familiarize yourself with any code it contains. The comments will provide a structure for you to follow.**

- Initialize the Bridges object with this assignment number, your username, and your API key. (See the Bridges template on Moodle for details)

- Observe the method calls building a List of Earthquake Tweets from Bridges, write a small loop to test the results if you like

- Create an instance of your Tree, and parameterize it to store EarthquakeUSGS objects (the same objects stored in the List from Bridges). Use the Earthquake's magnitude as the Tree's search key. You should also familiarize yourself with the methods each EarthquakeUSGS object contains

- Read every EarthquakeUSGS object from the List into your Tree, then use *Bridges.setDataStructure* to point Bridges to your data structure

- Create a menu in the console allowing users to specify operations and arguments. Each option should correspond to one of the new Tree methods you've added, and should visualize the tree after performing its operation.

**Deliverables –**

Your program should generate a number of visualizations (one for each user-specified command). Each successive visualization will appear as a sub-assignment on the Bridges website.

## Scoring Rubric

**Driver:** 40 points

- Up to 5 points for appropriate documentation and comments
- Up to 15 points for reading the USGS Earthquake Tweets into your BST
- Up to 20 points for creating a simple menu allowing users to specify operations and arguments

**BridgesBST:** 45 points

- Up to 5 points for appropriate documentation and comments
- Up to 5 points for implementing the **min** and **max** methods
- Up to 5 points for implementing the **reset** method
- Up to 10 points for implementing the **setLabel** method
- Up to 10 points for implementing the **findLocation** method
- Up to 10 points for implementing the **highlightRange** method

**Visualization:** 15 points

- Up to 15 points for visualizing the Tree after each operation. *Min, max, reset, setLabels, findLocation,* and *highlightRange* should all generate new visualizations.

**Total points available: 100**

**This will be graded as a programming assignment**

*Late programs will lose 10% of the available points per day.*