SIT221 –DATA STRUCTURES AND ALGORITHMS

LAB8: TREES

LAB OBJECTIVE:

The objective of this lab is to practice with Binary Search Trees (BST).

SUBMISSION INSTRUCTIONS

Please submit your work to Week08 assignment folder.

PREPARATION

1. Download the template project available in Week08 resources folder. The solution has two projects: **DataStructures_Algorithms & Runner** projects.
2. You can download the project and copy files in **Week08** in your current project. Also copy file **Runner08_Task1** to your runner project.
3. Find tasks that you need to complete.

LAB TASKS

BINARY SEARCH TREE

In this task we want to implement a Binary Search Tree and be able to traverse and display the data in the tree in Pre/Pos/In order.

1. In **DataStructures_Algorithms/Week08**, you are given two classes **BSTNode<T>** and **BinarySearchTree<T>**.
    a. **BSTNode<T>** class stores the information of a node in a binary search tree (BST). In particular, an instance of **BSTNode<T>** contains a value (of type T), and two references to the left and right child of that node.
    b. **BinarySearchTree<T>** class stores the information of a binary search tree by holding a reference called **root** (an instance of **BSTNode<T>** class).
2. In the **BinarySearchTree<T>** class, you are required to implement the **Add** method which adds an element of type T into a BST. Recall that the data

which is stored in a BST is maintained in order (i.e. the value of the left child <= the value of the right child).

3. In the **BinarySearchTree<T>** class, you are required to implement the **Search** method given an input element of type T. This method should return a **BSTNode** whose value is equal to the input element or null of that element cannot be found in the BST.
4. In the **BinarySearchTree<T>** class, you are required to implement the **Min** and **Max** method which return the min and max value contained in the BST.
5. In the **BinarySearchTree<T>** class, you are given the **Traverse** method that receives a traversal mode (defined in **TraversalMode.cs**) and textwriter to write the results to. The method calls other three methods to do traversal. You are also given the **PreOrder_Traversal** method (traversing a BST in the order: middle, left child, right child). Please read and understand both **Traverse** and **PreOrder_Traversal** method.
6. You are now required to implement.
   a. void InOrder_Traverse(BSTNode<T> node, TextWriter tw). Note that InOrder will traverse a BST in the order: left child, middle, right child
   b. void PostOrder_Traverse(BSTNode<T> node, TextWriter tw). Note that PostOrder will traverse a BST in the order: left child, right child, middle.

7. You are given **Runner08_Task1**. Run this code manually to get the results and then compare with what is generated by your computers. If we would like to represent our data in ascending order, what traversal mode should we choose?