

CSCI2010U – Laboratory #9

AVL Trees

Introduction

The goal of this lab is to implement insertion into a binary search tree (or a sorted binary tree). In one case you will be asked to not maintain the AVL condition during insert (as discussed in Lecture 19) and in the other case you will be asked to implement insert while maintaining the AVL condition.

To get started with the lab you can download the following files from the course website:

- **BTNode.java**: which implements a binary tree node
- **LinkedBinaryTree.java**: which implements a binary tree
- **Product.java**: which implements a produce, which we want to store in the binary tree
- **ProductComparator.java**: which implements a scheme to compare the values of two products
- **Driver.java**: which implements the primary driver for this lab

Activity 1: Implement the missing methods (10 marks)

Implement the three missing methods in `LinkedBinaryTree.java`:

- The first method (**insert**) inserts `Product` objects in the binary tree while maintaining the sort condition but not maintaining the AVL condition.
- The second method (**insert_avl**) inserts a `Product` objects into the binary tree while maintaining the AVL condition.
- The last method (**print_dot**) prints out the contents of the binary tree in dot format.

Please note that your methods should work correctly irrespective of the actual insertion order of `Product` objects into the binary tree.

What needs to be submitted?

Please submit the following Java source files on Blackboard:

- `BTNode.java`
- `LinkedBinaryTree.java`
- `Product.java`
- `ProductComparator.java`
- `Driver.java`

You do not need to submit your Eclipse project file or class files.