1 Introduction & Instructions

This lab is another timing analysis, this time between the global tree balancing algorithm (DSW) and the balance as you go AVL the algorithm.

When you are finished submit all your work through the MyClasses page for this class. Create a directory called Lab05, put each programming exercise into its own subdirectory of this directory, zip the entire Lab05 directory up into the file Lab05.zip, and then submit this zip file to Lab #5.

Make sure that you include a makefile, check the contents of the zip file before uploading it, and that the file was submitted correctly to MyCLasses.

2 Exercise

1. Create a single program that incorporates the following.
   (a) The templated binary tree class from the example code that has the DSW algorithm implemented. You will want the iterative version since these trees will be large and as we know the recursive version will have difficulty with processing the backbone.
   (b) The templated AVL tree class from the example code.
   (c) In the main, create two binary trees and an AVL tree that stores integers.
   (d) Let the user select:
      i. The number of nodes to insert into the tree (call it n).
      ii. The range of integers values to insert, the interval $[0, m)$.
      iii. A positive integer b, whose use is discussed below.
   (e) Set up a timing structure for each that will time,
      i. The insertion of the n random numbers into the AVL tree.
      ii. The insertion of the n random numbers into one of the binary trees and then a single balance at the end of the insertions.
      iii. The insertion of the n random numbers into the second binary tree with a balance after every b insertions and one at the end of all the insertions.
2. Run the timing analysis on different tree sizes and different element sizes to obtain some meaningful data. Compare and contrast the results and draw as many empirical conclusions as you can. Some of the obvious questions you will want to consider are the following, but don't limit yourselves to just these. As usual, write a report summary including your data, graphs, analysis, and conclusions. PDF the document and upload it with your code.
   (a) Is there a significant difference between the three methods?
   (b) Does the data size affect these three methods?
   (c) Does the size of the b insertion balancing value affect the DSW methods?
   (d) Does the graphing of these method timings with respect to n suggest anything about the complexity of these these algorithms?