COMP 410 - Spring 2017
Programming Assignment 4

Due back by **5:00 pm on March 31st**

For this assignment, you are to

1) implement the Dynamic Dictionary ADT as an AVL Tree, using *lazy delete* to implement the `remove` operation; and
2) compare the performance of this implementation to that of a (provided) Binary Search Tree implementation on various kinds of input data.

The Dynamic Dictionary interface that you are to implement, as well as a Binary Search Tree implementation of this interface, are available off the course web-page as a class handout for L14 (2017/03/01). You will need to

1) Write the class `public class AVLtree<K extends Comparable<? super K>, D> { ...}` that represents a single node of an AVL tree
2) Use this class to complete the class implementing the DD interface as an AVL tree, that is available for download off the course web-page as a class handout for L18 (2017/03/22). *Do not write any additional private methods* in completing the class.
3) Compare the performance of the "regular" BST and the AVL tree implementations on various kinds of test-cases – sorted, random, with and without various forms of remove operations interspersed amongst inserts. In these comparisons, the metric of consideration should be the height of the trees (this is why the public `height` method, which is not part of the DD interface, is defined.)

Submission instructions. You should upload the following three files in a `.zip`file to Sakai

1) A file titled `AVLTree.java`, containing your `AVLtree` class;
2) A file titled `DDasAVL.java`, containing your AVL implementation of the dynamic dictionary interface; and
3) A pdf file describing the experiments you conducted to compare your implementation with the provided BST implementation, your observations in tabular form, and any high-level conclusions you are able to draw based upon your experiments.