Assignment 5

CSE 326: Data Structures and Algorithms

Autumn Quarter, 1999, University of Washington.

Steven Tanimoto, Instructor

Due Friday, November 12, 5:00 PM, via turnin.

Perform this assignment either individually or in groups of two. Groups of two are required to turn in all parts, whereas individuals need only perform 1 and 2. Groups will receive 5 percent extra credit for completing the assignment. Individuals who wish to do not only 1 and 2, but also 3, 4, and 5 outside of a group can earn 8 percent extra credit by doing all parts.

1. (required by all) Implement the Delete operation for AVL trees. You are welcome to use the AVL tree code for representing and building AVL trees that has been provided by the instructor.

2. (required by all) Demonstrate that your code works by first constructing a tree containing the keys 0, 1, ..., 14, and then deleting the keys in the order 0, 2, 4, ..., 14, 1, 3, ..., 13. Print out the tree after each deletion.

3. (required by groups) Modify the Binary Search Tree and AVL code so that the keys are strings (representing words) and the info field represents a frequency value – the number of occurrences of that word in a document. Using the Lincoln Gettysburg Address as the sample document, build a tree of words and frequency values. Print out the tree in a neat way that shows the word for each node and the count that goes with it.

4. (required by groups) Instrument the code so that you count the number of string comparison operations involved in each find, insert, etc. Compare the performance (in terms of the number of string comparisons) of straight Binary Search Tree insertion to build the unbalanced Gettysburg Address tree against the performance of AVL tree insertion.

5. (required by groups) Perform a similar comparison using the large text file provided for Assignment 3.