Cosc 2P03 Fall 2014
Assignment#4

(Due date for assignment is Friday Nov. 7th 4:00 p.m., Late date Monday Nov. 11th, 4:00 p.m.)

Data Input Below.

Objective

To implement a top down splay tree.

The Assignment

Implement a BST which has a String element and thus represents the Key as well; add an integer variable C which will represent the number of occurrences of the element. Your BST should support the following operations. Insert, Delete, Find and InOrder. In the event of like elements you are to increment the count C field.

The BST is to be implemented using top down splay operations. Recall from class that the splay rotations are very similar to the AVL rotations. Thus, you should be able to borrow the double and single rotations which you have implemented in A3. In addition, you will need to implement the double single left and right which the splay operations require.

Load the BST with the supplied data using the top down insert splay. Print a SOT of this data (String element and count).

Rescan the input data and remove every element from the tree which starts with a letter between (r - t or R - T inclusive). Print a SOT to verify that your tree is still in one piece and BST compliant.

Data Input Above.

The data for this assignment is the text between but not including the headings Data Input Below and Data Input Above. You are to treat each word as case sensitive. You may ignore punctuation, brackets, white space and hyphens. A word can be a number. All words are to be trimmed.

Output expected:

- SOT listing and Pre-Order listing of the tree after it has been built.
- SOT listing and Pre-Order listing of the tree after the elements as described have been removed.

Suggestions

- There are many similarities between A3 and A4. You will be able to reuse, the parser (break apart the input in to words). You may be tempted to copy A3 to A4 and strip out those components which do not apply. This will give you a good start.
- The rotation methods you have implemented in A3 should be very handy, code up methods for the

double singles (DS), hint: a DS to the right is nothing but 2 calls to a single right SR.
- Write the insert method, this should be the hard part. Remember this is an iterative routine. No recursion other then the SOT.
- The delete should make use of the Find method. Follow the recipe as described in class. When you delete 1 occurrence of a key, the find should rotate that key to the root, then decrement the count. If the count reaches 0, you remove the node as described in class.

Verify that your algorithms for insert and delete give the correct results. Use a small test data set that you control. Try drawing the trees by hand for these small test sets. Once you have confidence that it works on the small set then graduate to the larger data set.

Submission Requirements:

- **Cover Sheet** completely filled out, available from: "http://www.cosc.brocku.ca/coverpage" Note: your assignment will not be marked unless one is submitted with the assignment on the assignment due date.
- **Commented and properly documented** source code listing, use Java Doc style.
- Listing of any input you used to test your program.
- Listing of your output which reflects the input.
- Source code is to be Java.
- Electronic submission, run the script "submit2p03" from sandcastle.
- Neatly edited response to each part. If you use any web or book sources be sure to reference these.
- Statement on coversheet with following information.
  - Platform, e.g. Mac, PC, Commodor 64, my Java enabled wrist watch.
  - Compiler Version, e.g. Java 1.6, Java 1.7 e.g.

Good Luck