Homework 5

Homework 5 is due at 9AM on 10/7.

In this assignment, you will implement a word-count program.

Your program should begin by asking the user for a local file to read. (You may want to copy this file to a local file for testing purposes: http://www.gutenberg.org/ebooks/1342.txt.utf-8).

As your program reads in words, it will remove all non-alphabetic characters (including punctuation and digits) and change any capitalized character into lower-case format. A words is defined as a sequence of characters separated by white spaces and new lines.

Thus, "Darcy's" becomes the word "darcys".

This sentence fragment (from near the end of chapter 4 of the above file):

Mrs. Hurst and her sister allowed it to be so--but still they admired her

becomes the sequence of words

- mrs (period and capitalization removed)
- hurst (capitalization removed)
- and
- her
- sister
- allowed
- it
- to
- be
- sobut (note that the dash marks are removed, so this becomes one word)
- still
- they
- admired
- her

You should define a new class BST that implements a binary search tree. Please design and implement the following methods (we will start doing this together in lab). You can include additional methods if you think it is appropriate:

- __init__
- size (giving the number of entries in the tree)

- height (giving the depth of the tree)
- find(word) - this performs a binary search to see if the word is in the tree, if so, returns the number of times the word appears in the input text file
- add(word) -- if the word is not already in the tree, this adds it with an initial count of one. If the word is present, this increments it by one

find(word) and add(word) should be fully recursive functions. Each node in the tree may keep track of the depth of the current subtree and the number of entries below it

After the work is loaded into the dictionary, the program should run a main loop that takes as input a test word, and finds if it is in the BST. The format should look like this

```
Query?  elizabeth
The word elizabeth appears XXX times in the tree
```

If the key word "`stats`" is given, the program should print out the number of entries in the tree and the maximum depth of the tree.

If the key word "`terminate`" is given, the program should terminate.
Answer the following questions and include the answers in your testing section:

(1) For the file http://www.gutenberg.org/ebooks/1342.txt.utf-8, what is the depth of your tree? What does that say about the number of operations to find a word?
(2) What would happen if the input to your program were sorted (as it was in HW 2)?
(3) What are applications for binary search tree? In what ways are they superior to lists? In what ways are they inferior to lists?
(4) Did you implement the extra credit (listed below)? If so please explain your testing strategy on the extra credit.

Extra credit: For extra credit, modify "add(word)" so that as it adds words to the tree, it keeps the tree balanced through tree rotation, so that every leaf node differs by at most one in depth. See http://en.wikipedia.org/wiki/Tree_rotation for details.

Please extensively test your assignment. When it is complete, put it in a script named hw5.<lastname>.py. Write a document (acceptable formats include pdf and text files) explaining how you tested your program (what test cases and strategies did you use) and the answers to the above three questions as hw5-

test.<lastname>.txt or hw5-test.<lastname>.pdf. Upload these files using the file upload tool available at https://www.ischool.berkeley.edu/uploader/?s=i206 Login with your ISchool userid and password and follow the directions.