

# CISC 320 010 Introduction to Algorithms (Fall 2005)

## Homework 3

Handed out: October 6, 2005

**Due date: November 3, 2005**

In this assignment, you are asked to implement a spell checker using the red-black tree.

A dictionary is provided at <http://www.cis.udel.edu/~lliao/cis320f05/dictionary>. Read in the dictionary into the red-black tree. Then, read in a document for spell check. For each word read, see if it is in the red-black tree. If it isn't, then output the word. This word is either mis-spelled, or not in the dictionary.

You are allowed to use as a template the code implementing binary search trees from any source, but you have to code the part for red-black trees (e.g., rotations and coloring) yourself. You only need to implement *search* and *insert*, not *delete*.

Include an option to count the number of key (string) comparisons, the number of color flips, the number of rotations, and the tree height.

Your implementation should compile and run under Unix.

Experiment 1: Populate the red-black tree with the dictionary and report the number of key (string) comparisons, the number of color flips, and the number of rotations. During populating the tree, count the tree height for every 100 inserts. Plot a curve of the tree height versus the number of inserts.

Experiment 2: Repeat Experiment 1 with a randomized version of the dictionary (available at <http://www.cis.udel.edu/~lliao/cis320f05/dictionary.rand>).

Experiment 3: Read in a sample document (<http://www.cis.udel.edu/~lliao/cis320f05/sample.txt>) for spelling check. Record the running time and compare it to that of an array based spell checker.

Please email your code as attachment to both the instructor and TA. If you have more than one file, make a single tar file and name it as `cis320_xxxx_hw3.tar`, where `xxxx` is your class ID. You should also bring in a hard copy of code listings and analysis to hand in at the beginning of class as usual. This should be a 1-2 page document with some plots and a brief discussion of how you did your analysis and what you learned.

### Grading:

- |     |   |
|-----|---|
| 15% | Technical issues: code compiles, easy to read, well-documented, etc.  |
| 55% | Successful implementation of red-black trees and the spelling checker |
| 30% | Testing and analysis  |

Note that it is possible to do a good job and get full credit for testing and analysis even if red-black trees and spelling checker are not completely successfully implemented.