

## Lab Assignment 3

### CSCI 311 Algorithms & Data Structures in C++

In your first lab you were tasked with counting the occurrences of words given an input stream. You were asked to use the STL map to help you sort and count words from the input stream. Typically the STL map is implemented with a Red/Black Tree.

In the third assignment you will write your own Red/Black tree that stores a `<string,int>` pair, where the string is a key and the int is a value associated with a given key. Remember a key is a unique word in the input stream and the key is the number of times it is present in the input file.

I would like for you to reuse as much code as possible. To shorten development time, please leverage your code from the first assignment and replace all code that used the STL map with references to your Red/Black tree implementation.

Your Red/Black tree will need to support insertion and you will also need to be able to print out the elements of your tree in order (Remember BST traversals.) You are not required to support deletion! Write a function to check the black height of a tree and use this to determine if your Red/Black tree is functioning correctly. Remember that we are manipulating pointers and be sure to test your code with small inputs to verify correctness.

For this assignment I want you to place your code for the assignment into several files. `filereader.h/cpp` will contain all of your code for reading the input. `wordcounts.cpp` will take formatted output from `filereader.h/cpp` and input words and counts into a Red/Black tree. All of the source code related to your Red/Black tree will be contained in `redblack.h/cpp`. The driver program `main.cpp` will contain all of the code to instantiate objects that are responsible for reading and counting words contained in the input file.

For your write up I would like for you to use a microsecond resolution timer and compare the run time of your red/black tree implementation with the run time of the STL map. If there are any differences between the two, provide an explanation of this behavior. In addition, include the black height of the children of the root of your tree for every single test case. Describe your overall design in your write up, include timing information, and include the black height of the two children of the root. Do not leave any code related to timing in your solution. Please describe the

overall design of your code. I will provide you with a Makefile that will be used to grade your solutions and use the input/output from lab1 to test your solutions.

This assignment is due Nov. 7, 11:59 pm.