Lab 5: Putting it all Together

By now you should have implementations of a list and a tree of your own as well as components for tokenization and for command line arguments.

We now need to put this all together.

Create a class called counter (in counter.cpp) that is going to manage the counting of words in each file. It should have a constructor that accepts the dictionary file and the text file that is to be loaded in. It should also accept the type of data structure that should be used with this file, as well as an output file to save the counts for each word to.

Have some decision logic that decided which data structure to use based on the arguments passed in. Load the dictionary into that data structure. Next, load each line from the input text file into that data structure. Don’t forget to handle any possible errors or failures that may occur.

Now, iterate over the lines read in from the text file and call your tokenization code to split each line into tokens. Search through your dictionary to see if this word is in the dictionary and if it is, update the counts for that word in your counts map.

Finally, write out the list of counts to the output file specified.