**Introduction**

This assignment builds a class to create, populate and search for items in different data structures then compare the execution time of each implementation. This time you will compare HashMaps and TreeMaps.

**Preparation**

See the announcements for the link to the GitHub repository for this assignment. You will need to create a new class to the project in your IDE and the repository as you complete this assignment.

**Directions**

Write a Java program to read a text file and count the number of times each word occurs. Use a TreeMap to do this. Words are anything separated by white space.

* Use a small data file for testing as you initially write your program. However, for the real time measurements, you need a large data file. Get a free book from [Project Gutenberg](http://www.gutenberg.org/)
* [Links to an external site.](http://www.gutenberg.org/). Make sure the .txt file is at least 200 Kbytes.
* Remove all the commas, periods, question marks and exclamation points. Do not remove any other punctuation. It will cause funky counts on small text files, but won't be a significant issue with large files.
* As you read from a file, store the words in a List (you choose). You will use this List for the remainder of the assignment. This avoids timing problems when reading the same file multiple times in the same program.
* All comparisons should be case insensitive, so that "Therefore" and "therefore" are counted as the same word.
* Start your timer, create a TreeMap, and then use the TreeMap to count the number of occurrences of each word in the List. After your program has counted all the words, print the 5 most frequently occurring words that are longer than 6 characters and their number of occurrences. Then stop your timer and record the result. Don't print any other words besides the top 5.
* Repeat the previous step but use a HashMap instead.  Record your measurements.

You get to decide on the structure of your program this time.

In a Word document, answer the following questions:

* Describe the state of your project, what works and what doesn’t.
* Describe how you tested your program, since the data file is very large
* Show your timing data and write a paragraph on the benefits/drawbacks of using a HashMap vs. a TreeMap
* In a sentence or two, what did you learn?
* In a sentence or two, what did you like about this project?
* In a sentence or two, what did you find confusing or would like to see done differently regarding this project?
* In a sentence or two, if you had another hour or two, what would you like to add to the project or how would you do things differently?

**Notes**

After your program is working for each step of the process, you should commit your changes to the repository. This allows you to go back to a previous working version in case you decide to throw out new code and try a different approach.

Note that finding the top 5 words will takes some extra effort. Whatever you do, don't do a sort and then take the last 5 items. Sorting is expensive for large data structures and not necessary for this exercise. Doing a sort is worth a maximum of 1 point.

**Submit**

Be sure to put your Word document AND your ***.txt book file*** in your repository and do one final commit. Generate a .zip file from the repository and submit that file to Canvas. Double check that what you submit to Canvas has all your .java files.

**Due Dates**

* by 11:59 p.m., Sunday, CT