Homework 1: Due: Fri. Feb. 3 by midnight via email to grader

The purpose of this assignment is to look at some well-known texts and find the most common words. In class we looked at a program that read lines from a file and broke it into words (admittedly not perfectly, it didn’t handle numbers or account for hyphenation). We also looked at integers and counted the maximum number of occurrence of an integer. You will put these two together, to find the most used word in a text of your choice, and how many distinct words were used. (If you have time, for extra credit you should improve the tokenizing algorithm, and include what you did in your email. How are capitalized words handled in the code?).

To make it more interesting, you should also use MergeSort instead of the selection sort we wrote in class, or the java.util.Arrays.sort() routine that I demonstrated. MergeSort will get us started at looking at recursive algorithms. It is also described in the textbook, so you have more than class discussion as a reference. Also, I don’t like the Weiss implementation. If you do use it, you should specialize it to Strings (since we haven’t covered generics yet), and you should improve the variable names and comments. (You might want to try writing your own before looking at it).

Make sure you debug your code on a small file before you try it on something like War and Peace. There are interesting test files of books you can find by going to http://www.gutenberg.org/ and clicking around.

This is a warmup assignment so it will be due in one week by email to the grader: dkc237@nyu.edu. Make sure your name is in the mail with your program attached, and that the subject line says HW #N. Note that the grader will run your code on his test files, so you should make sure your array is large enough, and the input doesn’t exceed the array bounds. For the purposes of this assignment, you can dimension your array at a million – we will be discussing data structures that are better solutions to this problem shortly.

The honors section should also add the following to the assignment Your program should include all three sorting routines mentioned above. Time them each using different size input files, (of a small enough sizes that it’s reasonable, say files with less than ten thousand words). Use the information to verify our expectations of running time. A plot may help here, along with some numerical calculations of running time as a function of input size N to back it up. Your homework should include a discussion justifying your conclusions about the running time and the plot.

To time a section of code you can call System.currentTimeMillis() before and after, and take the difference. Check the java libraries System class for more details.