CS 102 Lab No. 7 Fall 2019

Merge Sort & Text File I/O

Q1

Implement a linear-time static method called `mergeSorted` that merges two sorted `String` arrays, producing a new sorted `String` array. Assume that the input arrays are in ascending (non-decreasing) alphabetic order. For example, if the first input array is ["cats", "hogs"] and the second input array is ["dogs", "frogs"], then `mergeSorted` returns the array [“cats”, “dogs”, “frogs”, “hogs”]. Note that the inputs may be null or empty. If any input array is null or empty, then the output array should contain the elements from the non-empty one.

Hint: Keep an index into each array to process each element. Repeatedly append the smallest unprocessed element from either array and advance the index accordingly.

Q2

Using your `mergeSorted` method from Q1, implement the merge sort algorithm to sort a `String` array in ascending order.

Q3

Write a program that reads all of the words from an input text file into an array (download and use housman.txt file), sorts those words using your merge sort method from Q2, and then writes sorted words in a new text file. A word is defined as a sequence of uppercase and lowercase characters. The output text file must contain unique sorted words.

You should use the **`Scanner`** class to read words from a text file and **`PrintWriter`** class to write into a text file.

An excerpt from the expected output file for the input file housman.txt:

A
Although
And
Are
Battle
….
….
When
XIII
a
again
….
….
would
you
your