Assignment 2.3: Searching and Sorting

Due: 8am, Tuesday, November 25, 2008

Directions

The problems for this assignment are described below. You will submit this assignment by emailing it do Mr. Wulsin. The file will be named "Assignment 2.3-LastName.zip", where [LastName] is obviously your last name. All of your submitted assignments will be named in this fashion. The Zip file will contain a folder of the same name (so, "Assignment 2.3-LastName"), which will contain a folder for each of the subsequent problems. Each of these subfolders will have the name of its particular problem. Each of these problem subfolders will contain all of the necessary files for that particular problem. It's very important that you follow all of these naming conventions exactly as specified.

Problems

1) ZipCodes

In this assignment, you will read in an unsorted list of US zip codes and their corresponding city and state. You will create a class called `ZipCode` that contains `private String` data for the zip code, the city, and the state, all of which will initially be set in the `ZipCode`'s constructor. Make sure to also give this class getter methods to return the zip code (e.g. `getZipCode()`), city, and state values.

Once you write the `ZipData` class, you will finish writing the `ZipCodesData` class. The A track will finish writing the constructor, the methods for sequential and binary search, and the insertion sort method. The AB track will write all of these and also write the mergesort method.

The tester method, `ZipTester.java`, is already provided for you, but you will have to comment and/or uncomment different lines depending on what you are trying to do.

Another note, when comparing Strings, you will use the `.compareTo()` String method, to determine whether one string is "larger" or "smaller" than another. Look up exactly how the `.compareTo()` method works in your book or online.

You may find it useful to test your searching and sorting methods with a smaller number of zip codes (there are about 41,000 zip codes in the `UnsortedZipData.csv` file). Simply open the file in eclipse or Wordpad and then copy/paste a chunk of it into another file. Insertion sort of the whole file takes a minute or two, which you don't want to have to waste every time you test your Insertion sort method.

Finally, once you have all of your code working, test out the difference between the number of comparisons involved in sequential search O(n) and binary search O(log(n)). Test the difference between the number of comparisons involved in insertion sort ($O(n^2)$) and mergesort (O($nlog(n)$)).