**Introduction**

This assignment lets you read data from a file then sort it by two different methods. One is the built-in sort() method, and the other will be your own method that implements a mergeSort algorithm. You will compare the performance of the two implementations.

This assignment builds a set of classes to sort a linked list of objects in multiple ways and compare the execution time of each type of sort.

**Directions**

* Write a class WeatherReport, that has the following methods:

1. Constructor with no parameters. It should create a linked list of 8-10 Temperature objects which you can use for testing. Each Temperature object should have a City, State, Low Temperature and High Temperature, along with a method Differential, that returns the positive difference between the low and high temperature for that object. Don’t include any other attributes in your Temperature object.
2. Constructor with one String parameter – the name of a text file with weather information. The file has a header row, then many rows of temperature recordings. For example

Average,City,Code,Direction,Full,Maximum,Minimum,Month,Precipitation,WindSpeed,State,Day,Year  
39,Birmingham,BHM,33,1/3/2016,46,32,1,0,4.33,Alabama,3,2016  
39,Huntsville,HSV,32,1/3/2016,47,31,1,0,3.86,Alabama,3,2016

This constructor will create a linked list of Temperature objects, one for each row of data in the file. Your project will have the data file [weather.txt](https://canvas.park.edu/courses/74667/files/10204652?wrap=1)

[Download weather.txt](https://canvas.park.edu/courses/74667/files/10204652/download?download_frd=1) which you should use for the assignment.

1. isSortedByCity() – this boolean method returns true if the linked list is in sorted order by city, and false otherwise. Calling this method on the list generated by the constructor should return false.
2. isSortedByHigh() – this boolean method returns true if the linked list is in sorted order by highest temperature, and false otherwise. Calling this method on the list generated by the constructor should return false.
3. sortWithCollections(String by) – sorts the linked list using Collections.sort(). The parameter by tells you whether it should be by City or by high temperature.
4. sortWithMerge(String by) – sorts the linked list using your own merge sort implementation. The parameter by tells you whether it should be by City or by high temperature.
5. main – test your class by writing a main method that creates 4 WeatherReport objects with exactly the same data, and tests each to be sure it is not already sorted. Sort one by calling sortWithCollections(“City”) and the other by calling sortWithMerge(“City”). Be sure to time the results of each sort and record the times. Repeat this for sorting by high temperature. Be sure your linked lists really are sorted.
6. Answer the following questions in your Word document:
   * Describe the state of your project, what works and what doesn’t.
   * Describe how you tested your program
   * Show your timing data and write a paragraph on the benefits/drawbacks of writing a custom sort routine.
   * 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.

You might find the Java split() method in the String class helpful when processing data read from the file.

**Submit**

Be sure to put your Word document 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