The Report
A: Problem statement. In one or two sentences, summarize the goal of the problem.

The purpose of this assignment is to design a program that can read data from input files by using a linked list to find the max, min, median and print out the time complexity of each compilation.

B: Algorithm design. This project does not have a lot of algorithm design decisions. However, there are few areas that you will need to make a decision about how to compute the output values such as ‘min’, ‘max’, and ‘med’. Within one page, provide a brief description of the main ideas behind your algorithms for finding ‘min’, ‘max’, and ‘med’ within the list. Discuss any potential alternative approaches that could be used to find these values which could have resulted in a poor timing performance.

To approach this assignment, first I need to implement a linked list class. Inside the class, I created a insertionSort method which would take every input that is passed from the main, store it in a linked list then sort those values at the same time. While inserting and sorting the values, I keep track of the head node which would initially be the value of the min. After sorting the linked list, the last index will always be the max value of the linked list. Hence, to find the max value of the linked list, I created a while loop and found the value which was located next to the null node. The median was a bit challenging since we can not access the value from the linked list at a specific index directly. Therefore, I came up with an algorithm which I acquired from practicing Leetcode by using two nodes to traverse through the linked list. The ideas is that I would have the fast node which would travel 2 nodes and the slow node only go by one, and they both travel through the linked list at the same time. By the time the fast node reach the end of the linked list, the slow node would be at in the middle of that list since the fast node travels twice the length of the slow node.

A potential case that make this program poorly performance could be iterating through the list by using the for loop, increment a counter variable by one then divided it by half to find the middle value of the linked list.

C: Experimental setup. In this section, you should provide a description of your experiment setup, which includes but is not limited to:
● Machine specification - CPU maker, CPU speed, RAM available, hard drive type
CPU maker - Apple
CPU speed - 2.8x CPU performance, 5x graphics speed, 11x machine learning speed
RAM available - 201.54 GB available
Hard drive type - USB-C G-technology 1TB G-DRIVE
● How many times did you repeat each experiment before reporting the final timing statistics?

Around 5-6 times for each experiment

● O/S and environment used during testing: Windows or Unix? Also mention which compiler environment (e.g., javac version). This information will help the TAs determine where to run your programs during grading.

-Using Eclipse on a Unix system.

D: Experimental Results & Discussion

. In this section, you should report the performance (running time) and outputs based on your observations, and provide justification for your observations. For your testing and reporting, conduct the following two experiments using the two provided input files, namely ‘input1.txt’ and ‘input2.txt’.

input1.txt:
Min value: 1
Min time: 130583
Max value: 4000
Max time: 35042
med: 2057
med time: 23041

input2.txt:
Min value: 75
Min time: 1025958
Max value: 7999707
Max time: 1851875
med: 3998801
med time: 1219667

From my perspective I would say that the observation is well connected to the design of the program. Since the program keeps track of the head while inserting the new value, its time complexity should take the least time to compile. Because the program only travels half of the time through the total length of the linked list, the median time complexity should be faster than traverse till the end of the linked list to find the max value.