Eric Dazet

Nathik Salam

CPE 419-01

27 October 2015

Writeup: Vector Sum and Analysis

Part A

For this part of the assignment, we offloaded both the vector summation and histogram creation steps. Both used OpenMP and the vector summation used vectorization as well. Our program started scaling well, but with the larger files, the scaling dropped due the time it took to read in and write out the large vectors. This was also using the maximum number of threads, 224. The run times versus input size can be found below in Figure 1.

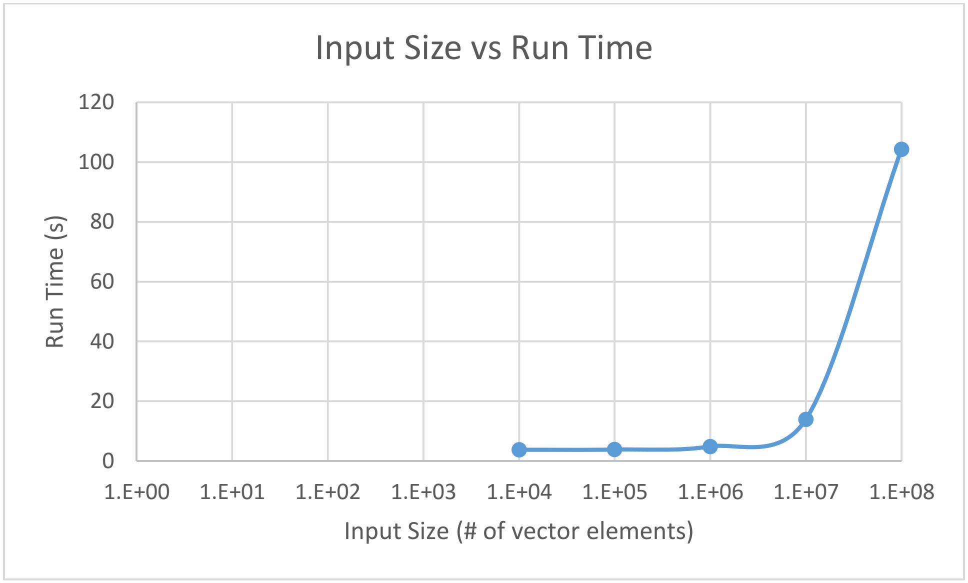

Figure 1: Input Size vs Run Time

For the parallel histogram creation, we used the following link as a starting point and then adapted it to work for offloading as well as for the specific bin scheme:

http://stackoverflow.com/questions/16789242/fill-histograms-array-reduction-in-parallel-with-openmp-without-using-a-critic

After implementing the parallel histogram function, we saw a decent speedup from our serial histogram function. The generated histograms for each input size can be found below in Figures 2 – 6. It is important to note that for the input vectors, hist.a and hist.b, the forty bins were spread across a range from -10.0 to 10.0. For the summation vector, hist.c, the same forty bins were spread across a range from -20.0 to 20.0.

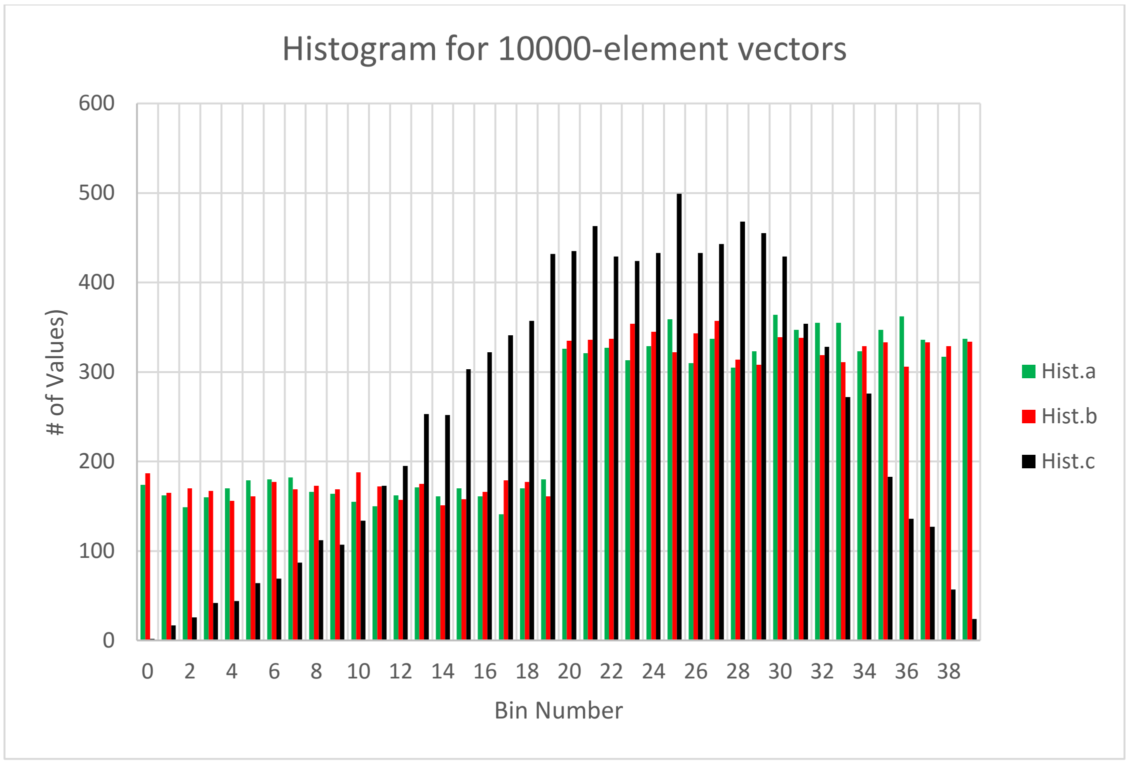

Figure 2: Histogram for 10000-element vectors

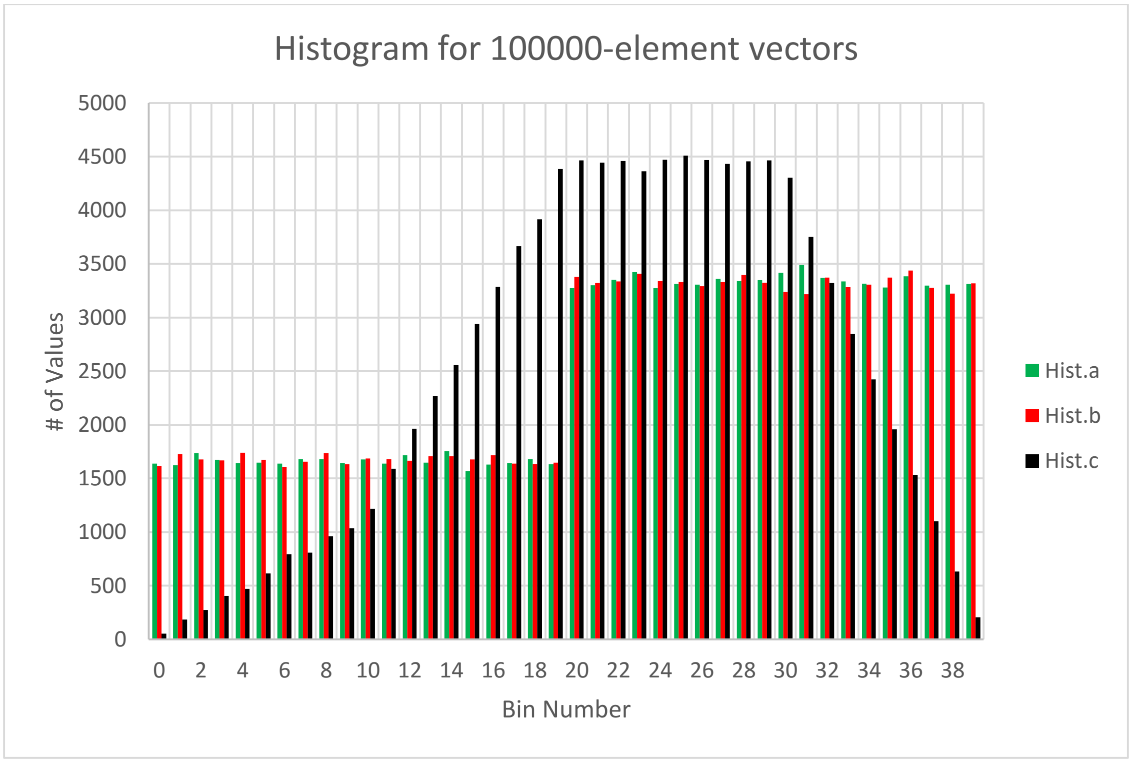

Figure 3: Histogram for 100000-element vectors

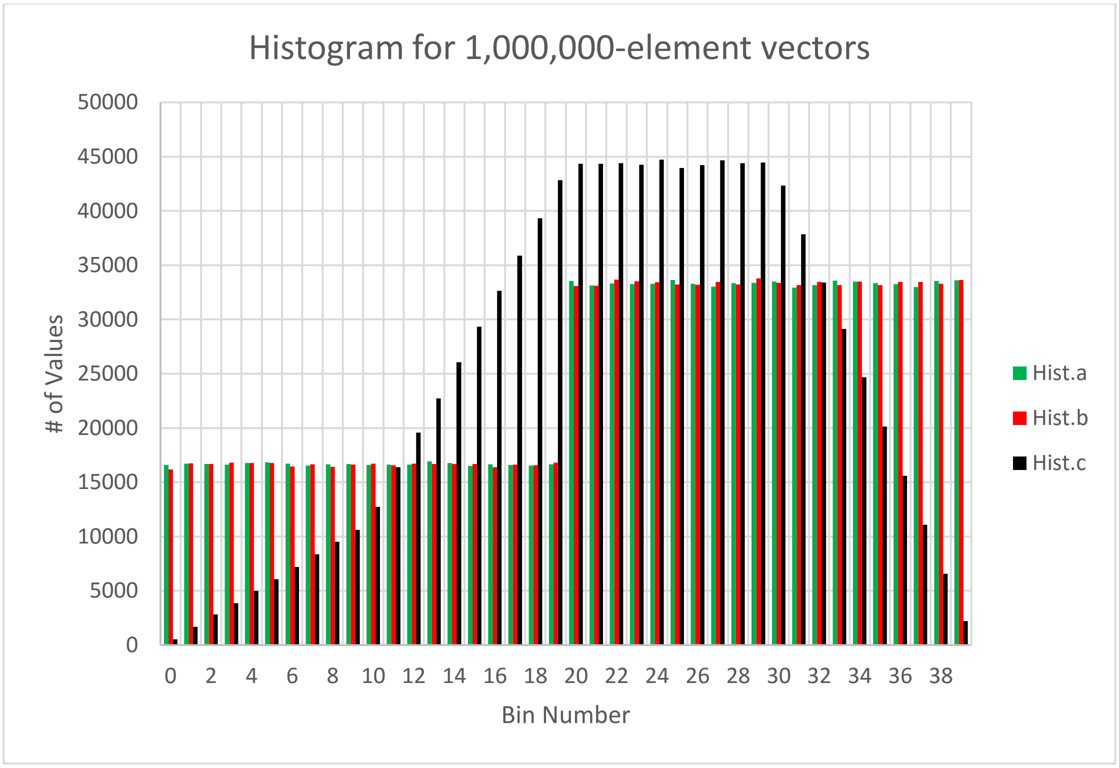

Figure 4: Histogram for 1,000,000-element vectors

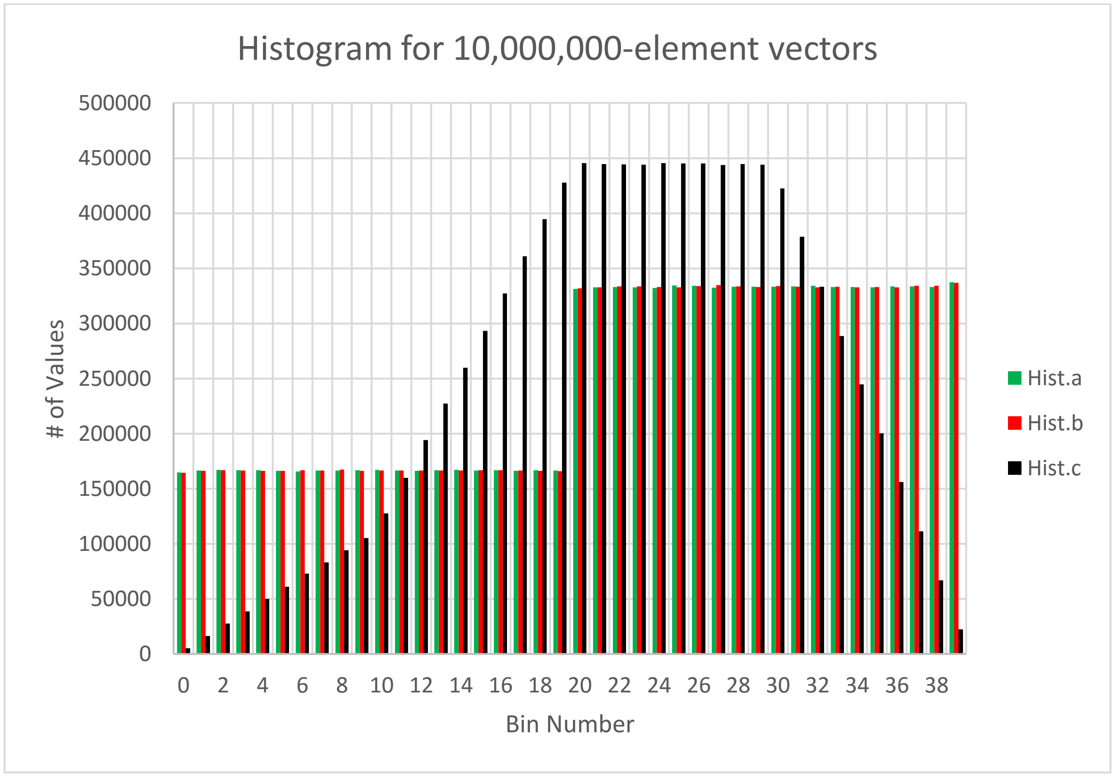

Figure 5: Histogram for 10,000,000-element vectors

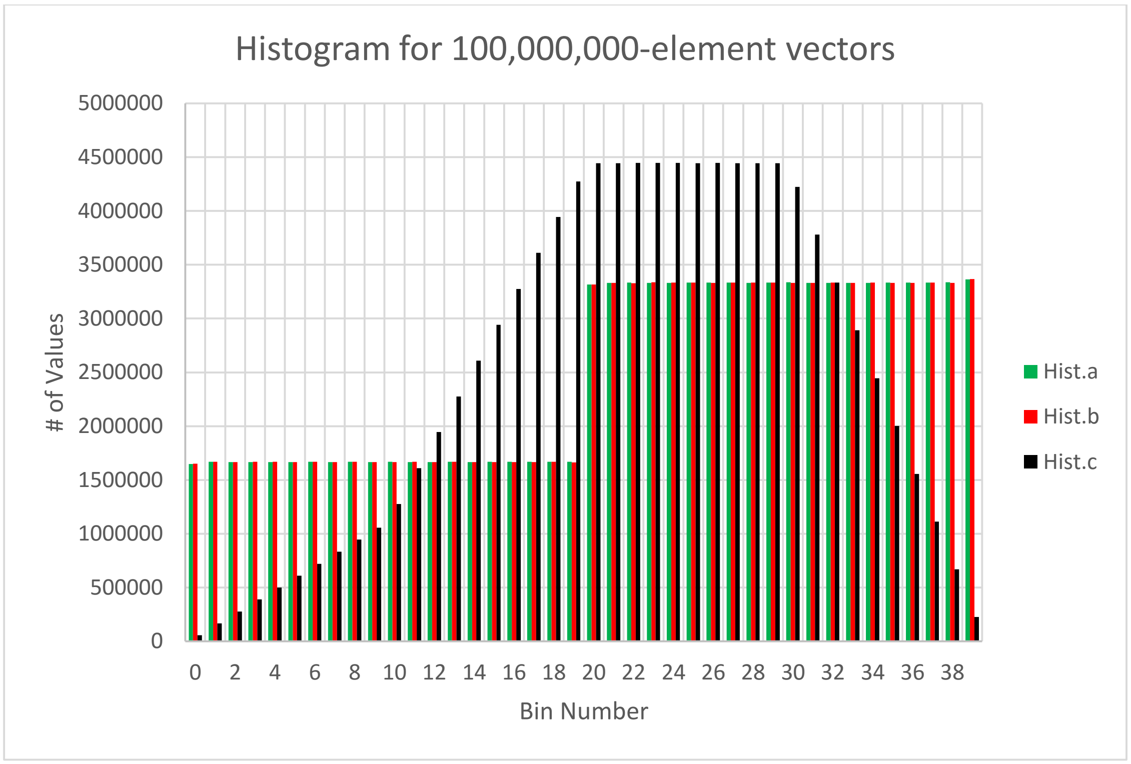

Figure 6: Histogram for 100,000,000-element vectors

Part B

For part B, we decided to use functions from Intel's MKL library. We were able to find a function that computed the mean and variation at the same time, which we would then use to solve for the standard deviation. Because of this, the time to calculate the mean and variation is a single value.

For minimum, maximum, and median, we first sorted the array and then simply indexed where appropriate.

For this part, we timed the following operations:

- Minimum
- Maximum
- Median
- Mean and Variation
- Sort
- Program Runtime

Below are graphs that plot the time it took to do each operation as a function of input size, found in Figures 7 – 12.

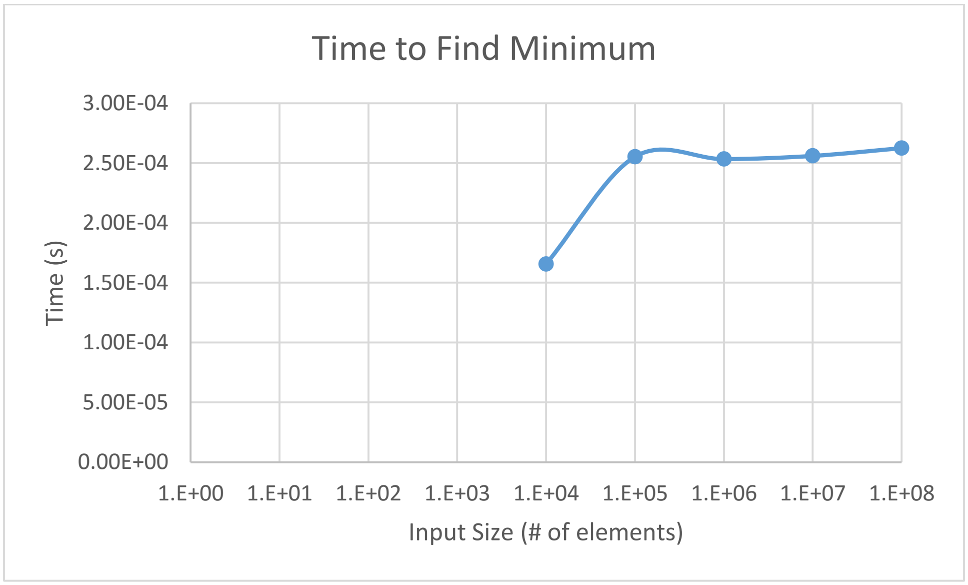

Figure 7: Time needed to find minimum

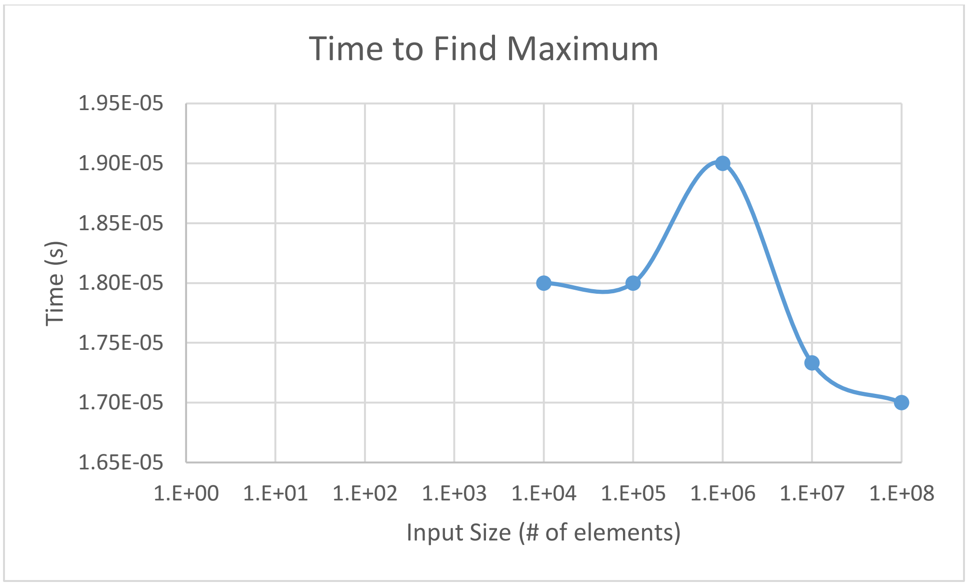

Figure 8: Time needed to find maximum

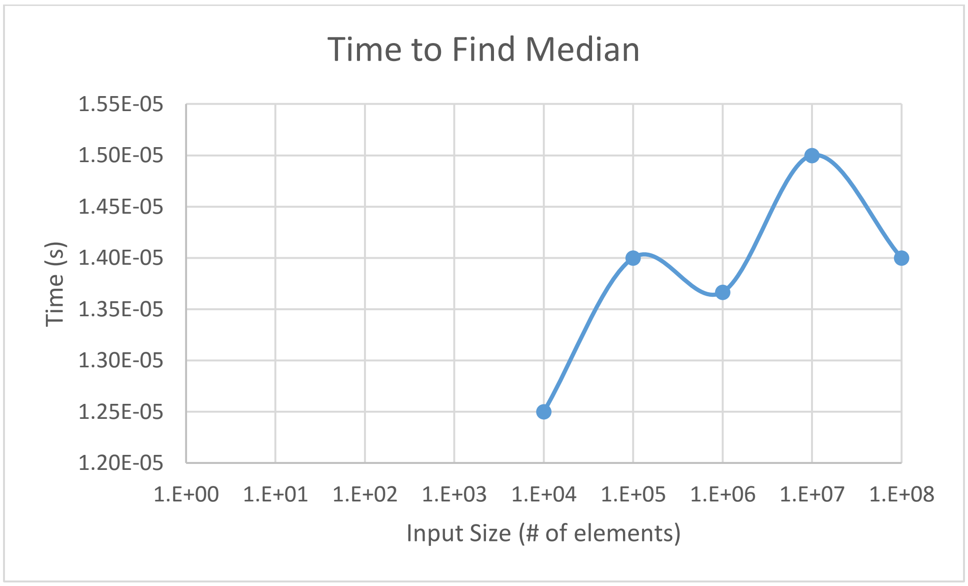

Figure 9: Time needed to find median

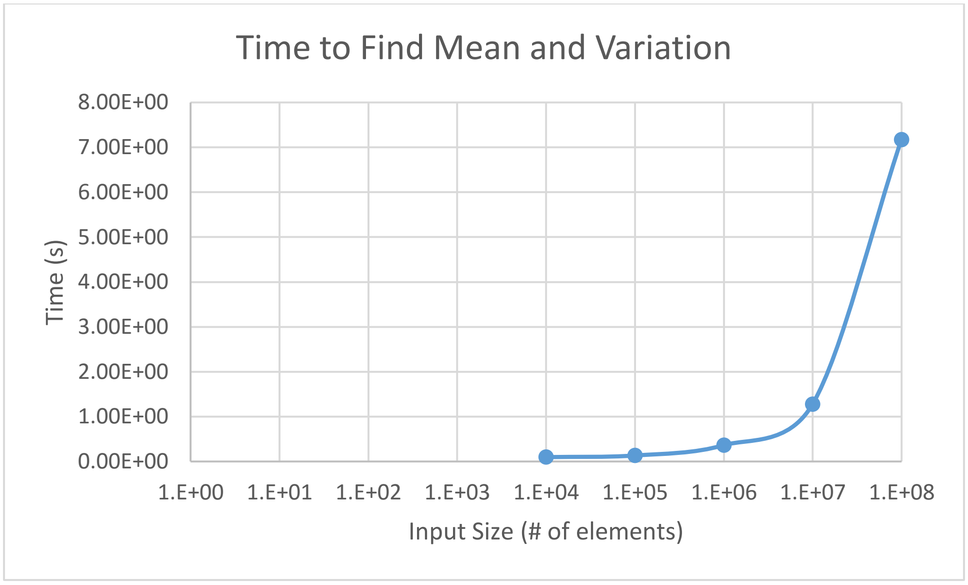

Figure 10: Time needed to find mean and variation

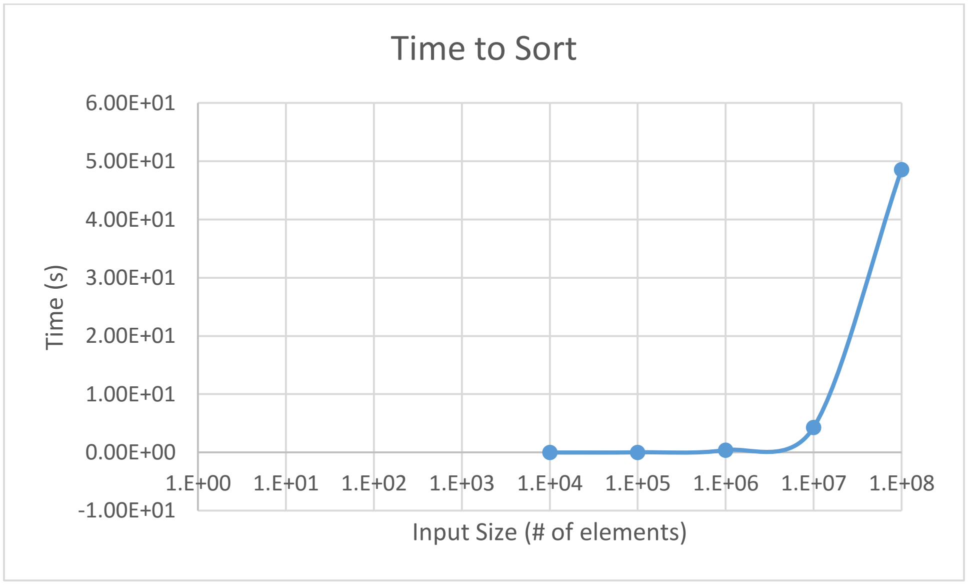

Figure 11: Time needed to sort

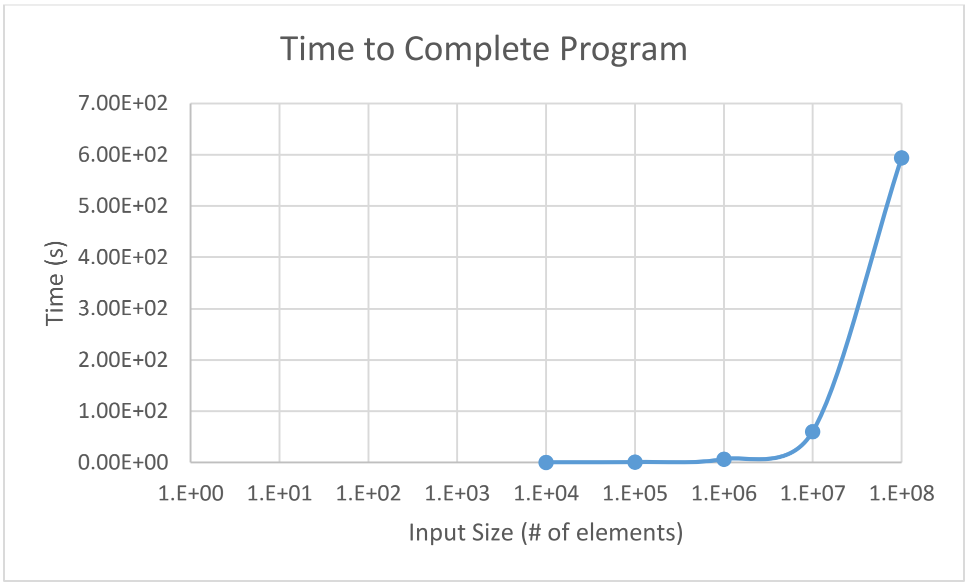

Figure 12: Time needed to complete program