**Spring 2022: The First Row**

**Team members:** Trent Menard, John Rempe, Hassan Khan

**Experiment Report**

**Introduction**

The purpose of this project entails calculating and recording the speed times of various sorting algorithms as well as comparing their sorting speeds to conclude what the two extremes are, the slowest and the fastest, but in a more presentable manner with the help of charts and graphs. We will also be able to highlight the sorting algorithms which fall between these two extremes with average and below to above average speeds.

The following are the algorithms this project has focused on, and as per the book, they have been categorized as fast or slow and the results from our code shall say the same:

*Fast Sort Algorithms*

* Quick sort
* Radix sort

*Slow sort Algorithms*

* Bubble sort
* Selection sort

Besides the above commonly known algorithms, we will also be presenting a new algorithm called TimSort that shall be faster, but confirmingly not slower than O(n log n).

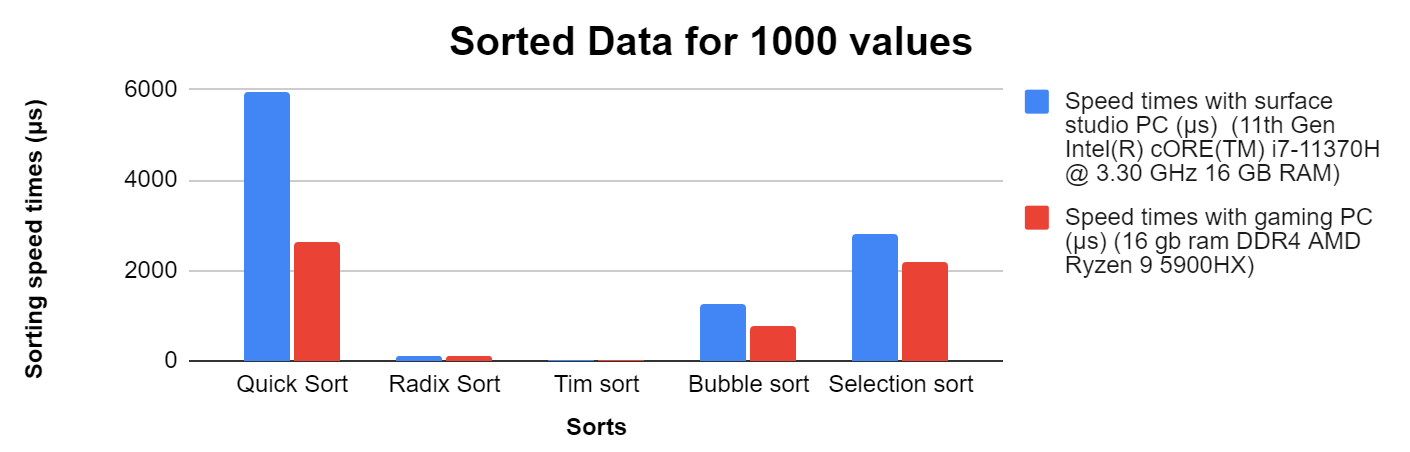
**Methods**

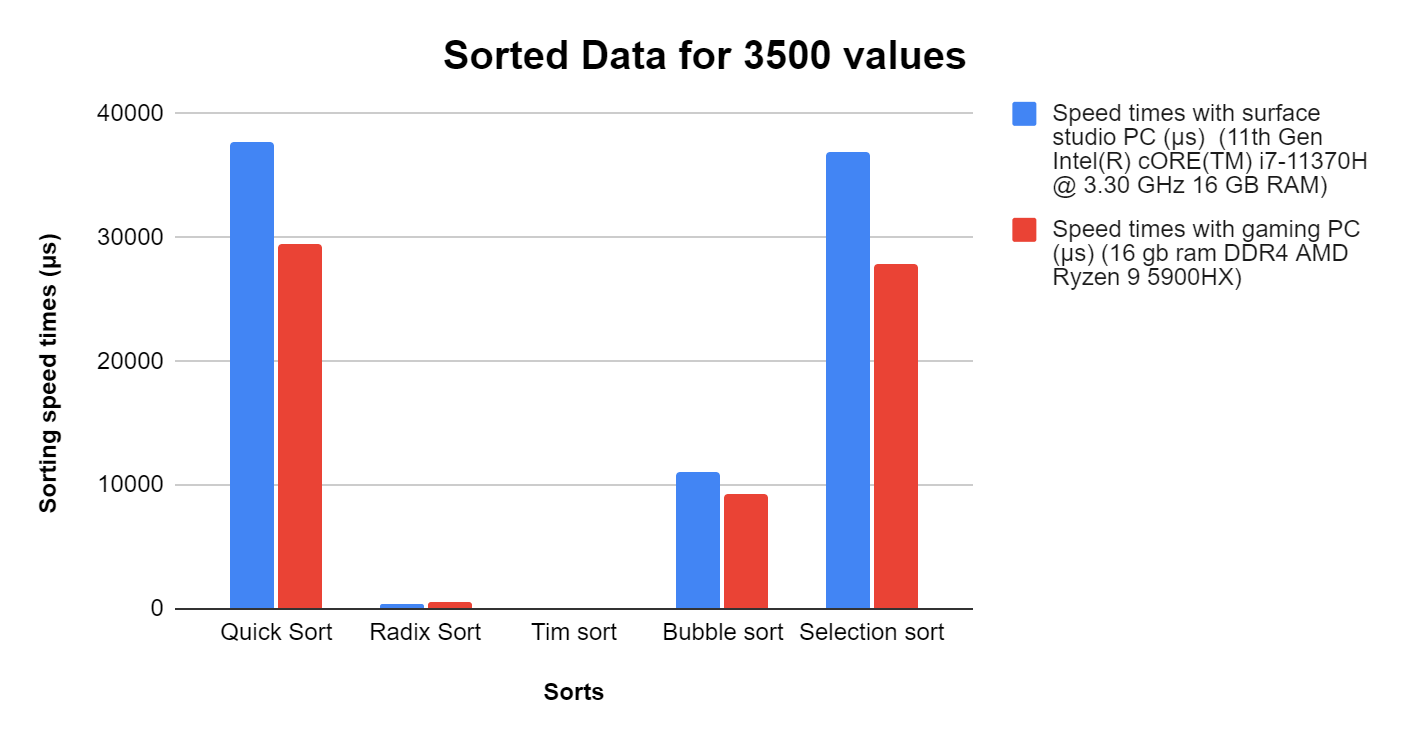
Modules were introduced in C++20. They essentially replace header files for easier API creation. They also speed up compilation time by not having to copy the header files into the .cpp every time. The primary interface, GenAlgs, is defined within a module.

See: <https://docs.microsoft.com/en-us/cpp/cpp/modules-cpp?view=msvc-170> for more information. The constructor makes a copy of the original data and then the sorts return a new array, so the original data is not modified *(by Trent Menard)*.

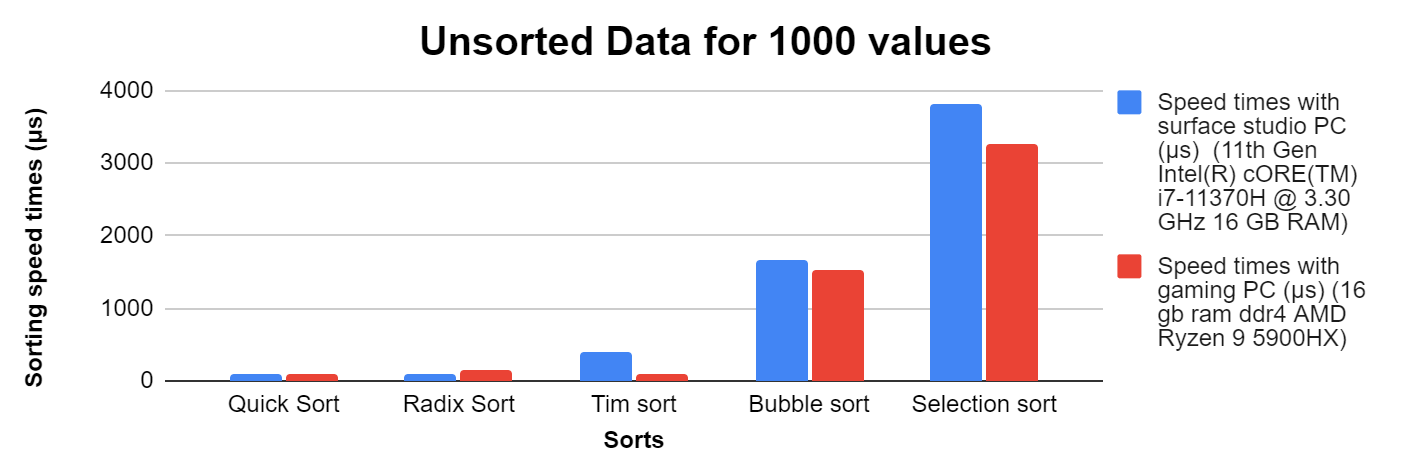
**Results**

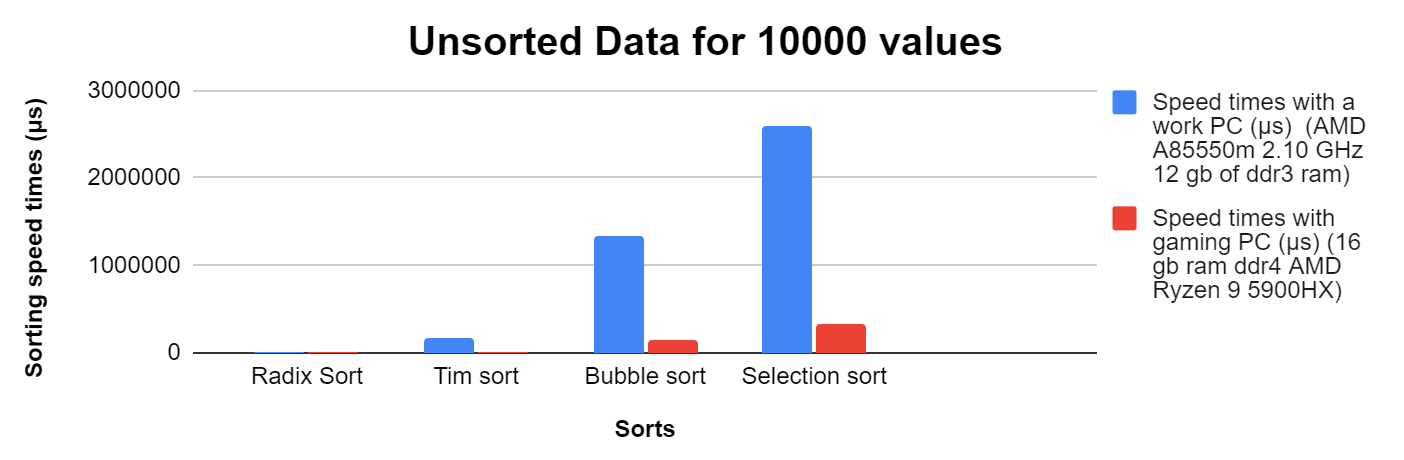
Bar charts for comparing each algorithms ***sorted*** (with link to spreadsheets):

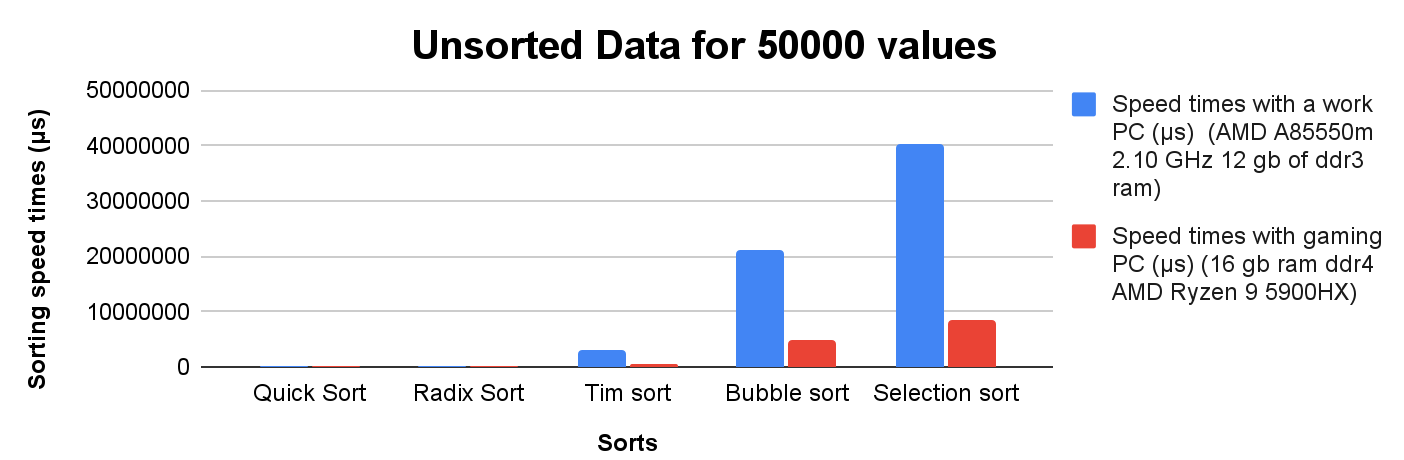


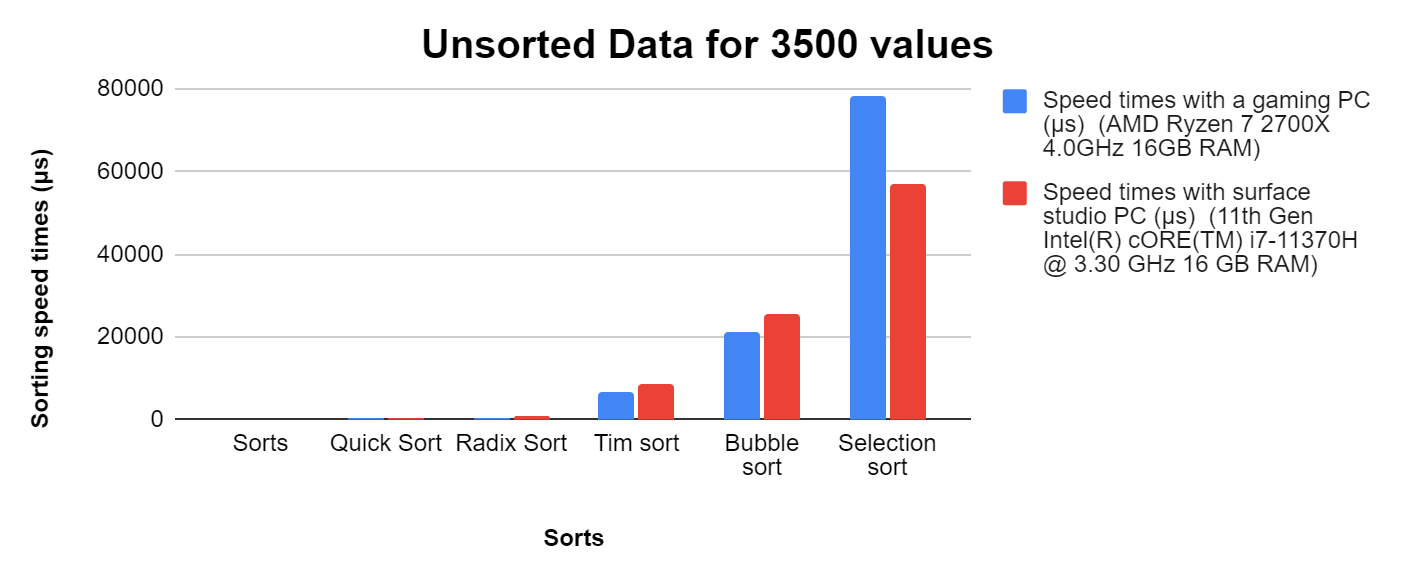


Bar charts for comparing each algorithms ***unsorted*** (with link to spreadsheets):









**Analysis/Discussion**

Challenge faced was for sorted data- the range of values that could be used, in our experience, was capped at 3500 for sorted data with Quick Sort. So the only ranges of values that were used for sorted data are 1000 and the maximum at 3500.