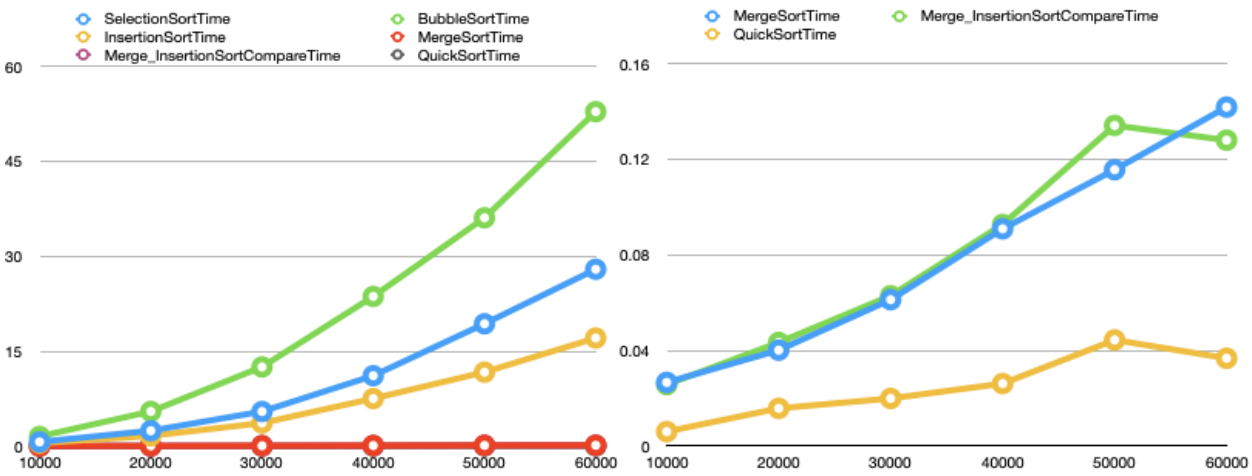
Jesus Ugarte
University of Central Florida
Computer Science 1

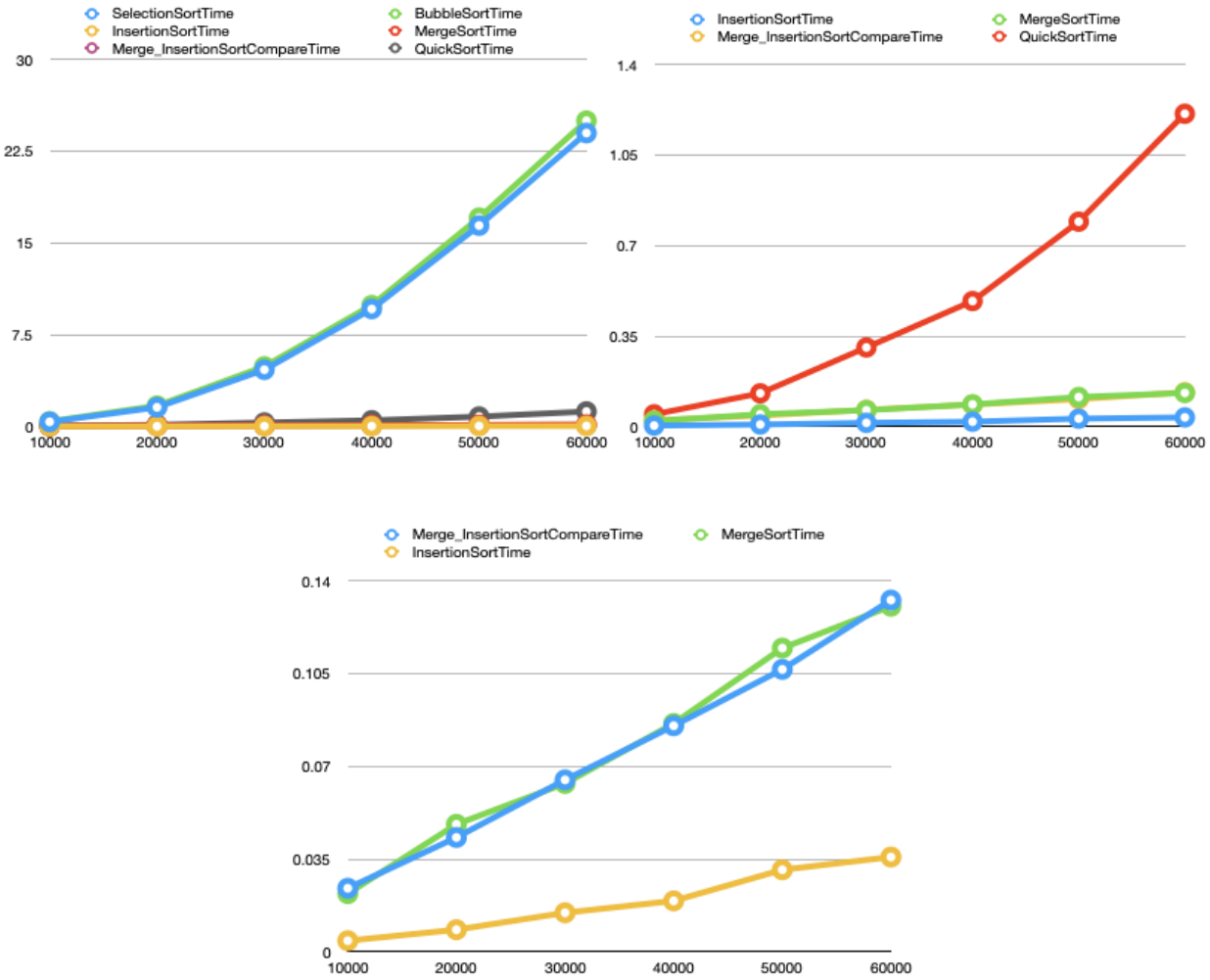
Sorting Algorithm analysis

Criteria 1 – Sort by name



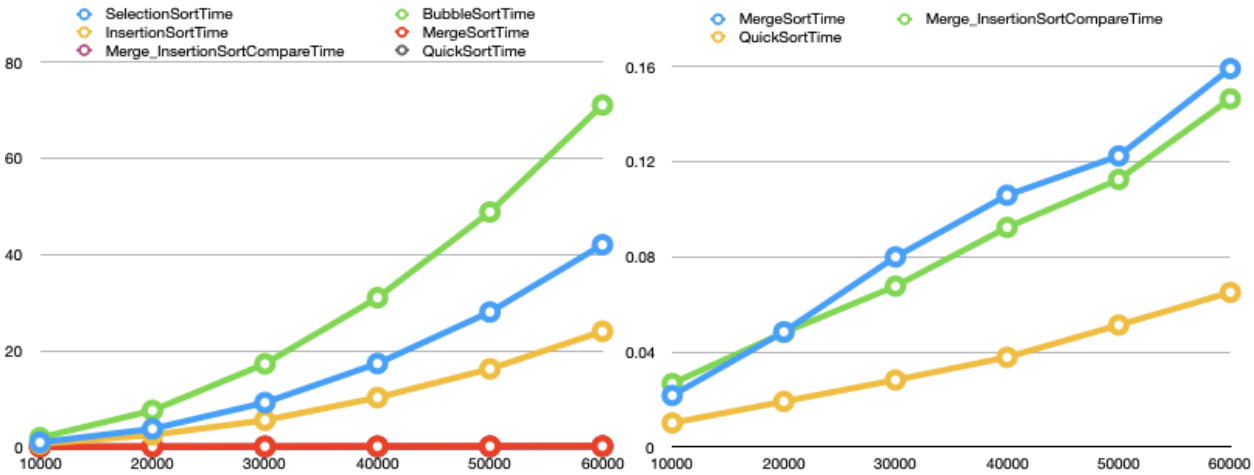
For the first criteria we can see that for bubble sort, selection sort and insertion sort, the run time grows exponentially with input file size. O(n^2) for all of them. Hence, we can expect them to take longer and longer for big chunks of data. For the merge sorts and quick sort, we can see that they overlap the scale, so I created a second graph to better appreciate how it looks like. We can see that quick sort takes less time overall. However, we can see a spike down in the 60K file because of the random pivot. Sometimes we can get lucky with the pivot and get better run times.

Criteria 2 – Sort by weight



For the second criteria, we can see that selection sort and bubble sort are very similar. Since we are handling with numbers, we tend to see pretty similar average times rather than with names. When dealing with numbers we can see that quicksort takes a little bit longer than insertion and merge sort. Hence, I can conclude that when sorting numbers, quicksort would take more and more time when compared to merge sort and insertion sort. Lastly in the last graph we can see that actually the fastest sorting algorithm for number sorting is insertion sort.

Criteria 3 – Sort by name and weight



For the last criteria, we can see that run time grows exponentially for bubble sort, selection sort and insertion sort. Quick sort, merge sort and fast merge sort overlap so I decided to include a second graph to better see the results. I am able to see that quicksort is actually the fastest sorting algorithm for name and weights, and merge sort would actually take closely similar time when comparing both.

Conclusion

I can conclude that depending on the sorting criteria, it would be beneficial to decide for a sorting algorithm so that we can sort the data more efficiently. Overall I was able to see that Bubble sort takes the longest in most of the cases, however, despite insertion sort takes long when sorting names it actually works better with sorting numbers, On the other hand quicksort is overall the best sorting algorithm in run time for name criteria and name and weight criteria. Also, it is very important to consider that each sorting algorithm takes different amounts of memory, so it is definitely an important factor to consider when deciding for a sorting algorithm. Merge sort takes a lot of memory, we could probably enhance it with the implementation of insertion sort when the split array is less than a certain number so that we could have a faster sorting algorithm.