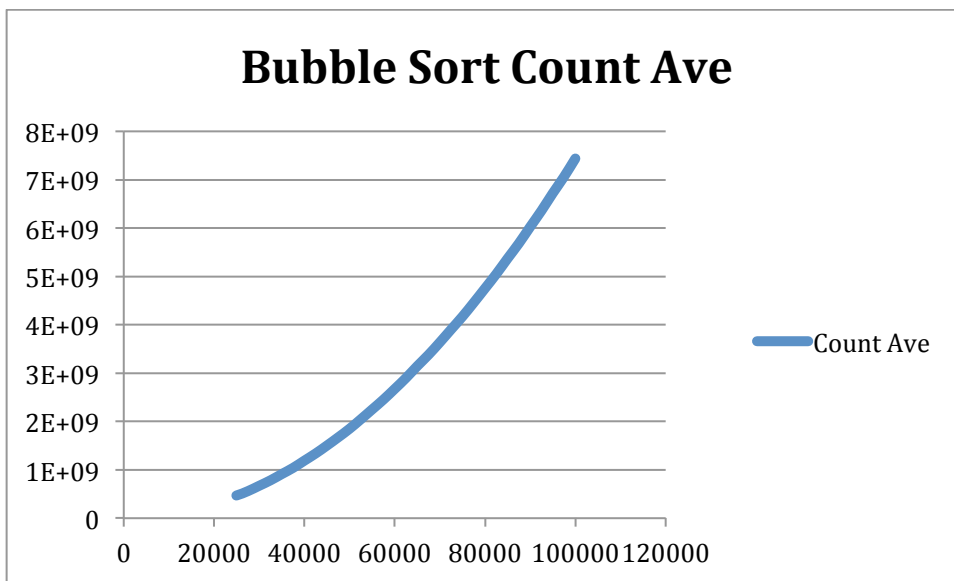
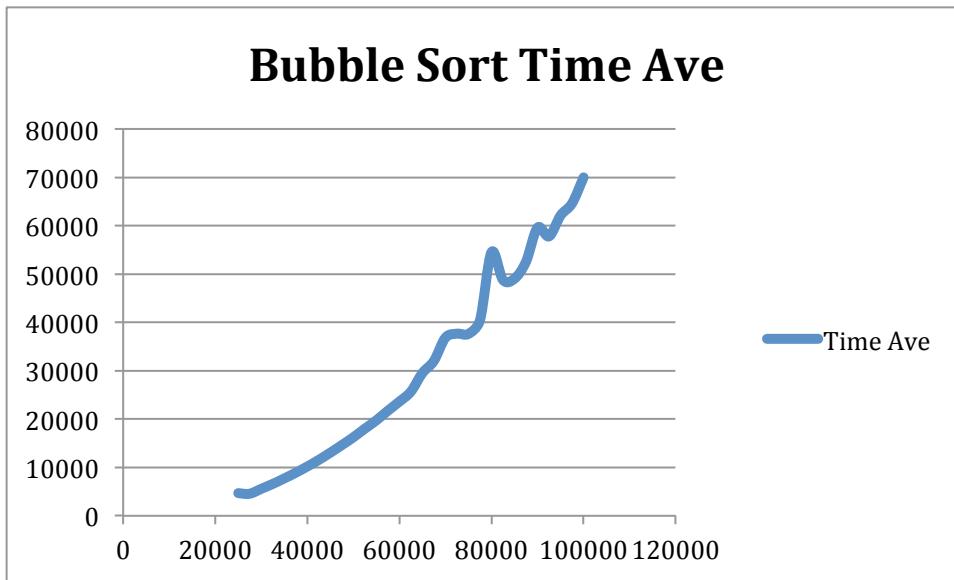


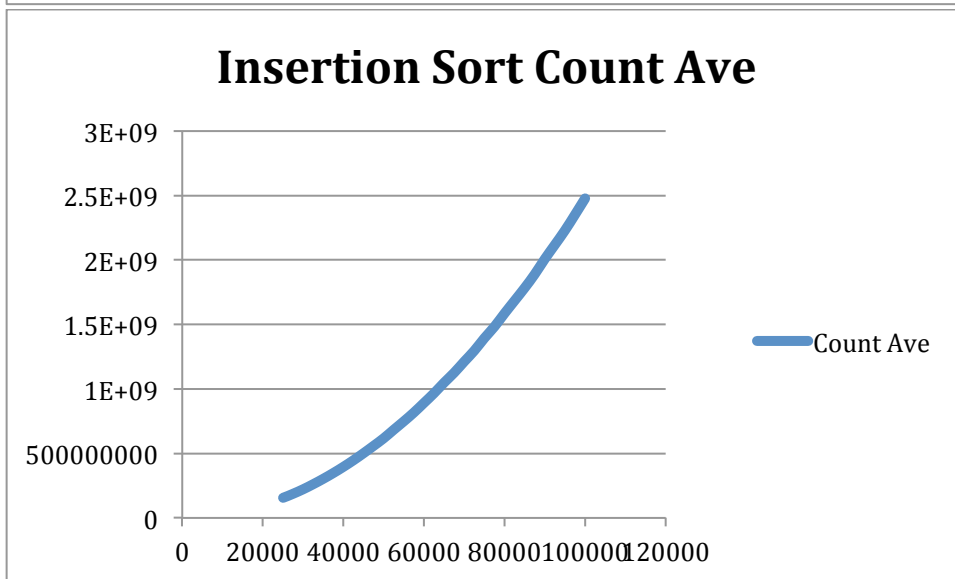
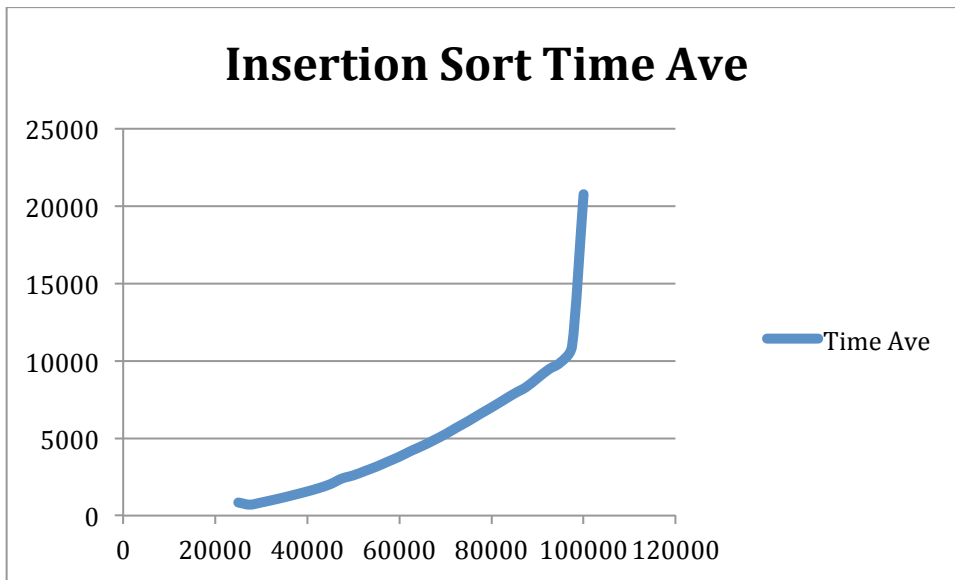
Borja Rojo
University of Rochester
CSC 172
Lab 14
Sorts

Bubble Sort



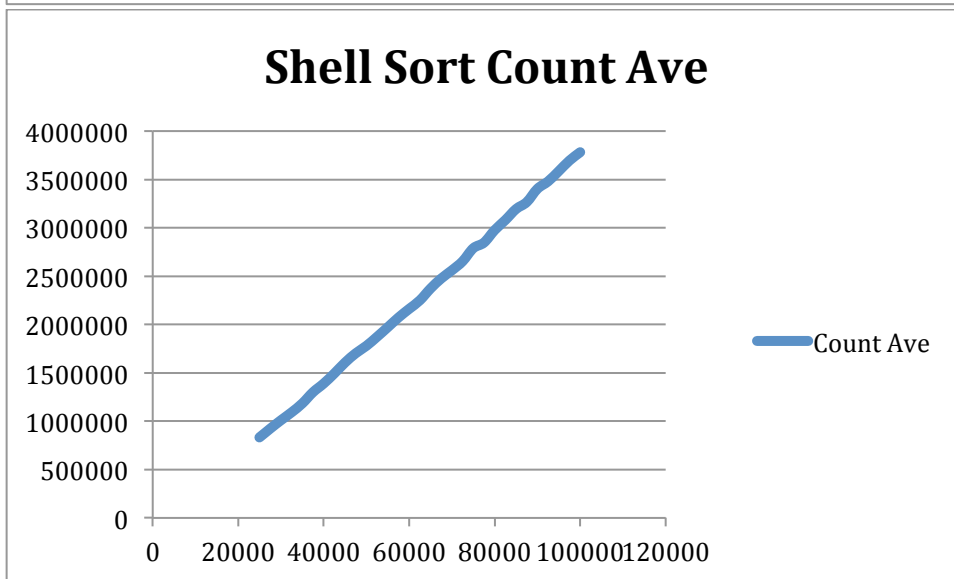
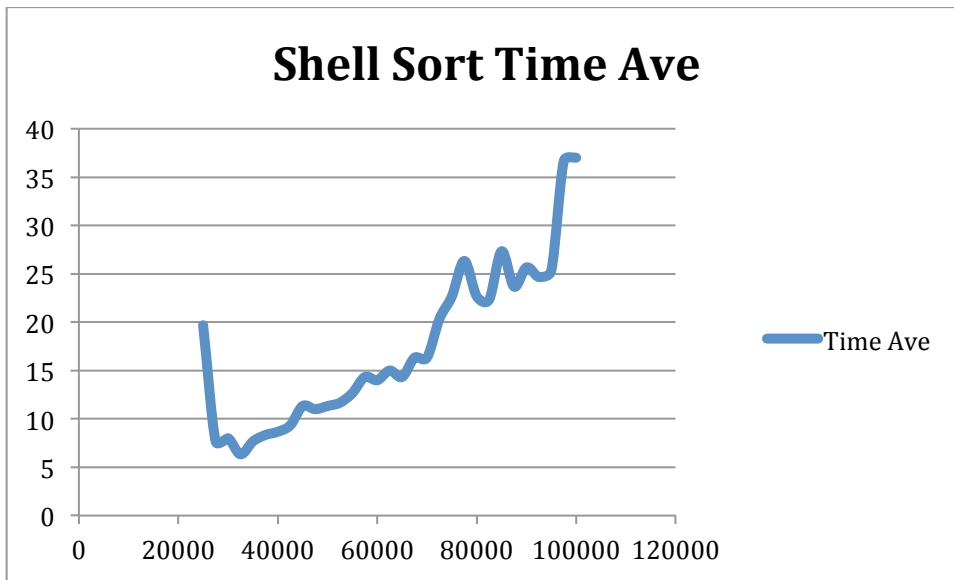
Runtime of $O(n^2)$

Insertion Sort



Runtime of $O(n^2)$

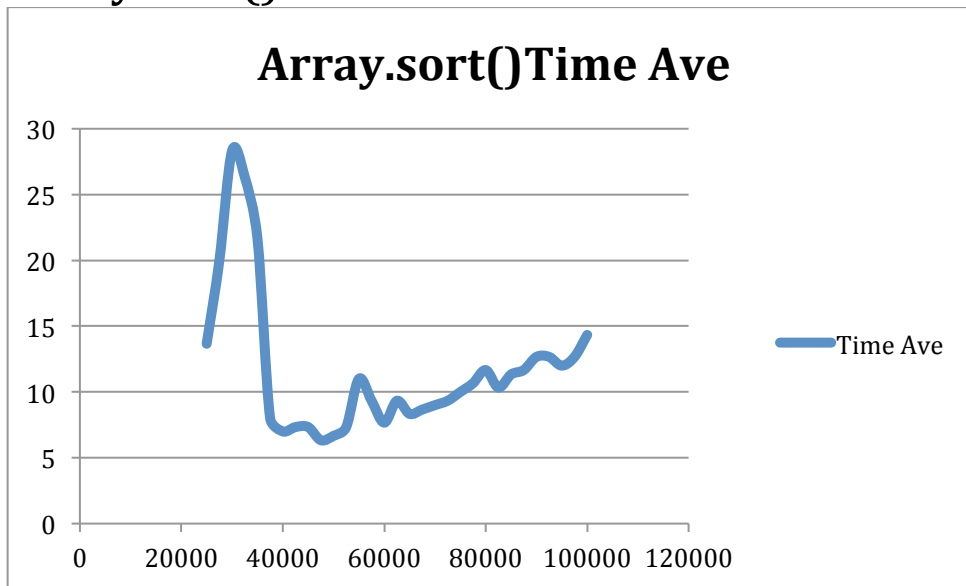
Shell Sort



Runtime of $O(n^{3/2})$

This is an interesting result, as the resulting data shows that the number of counts increases linearly and the time increases linearly for this portion of the data as well. There are some spikes in the average at the end, but there is a clear line that can be drawn under the Shell Sort Time Ave data that can hug the data at the lower points.

Array.sort()



Runtime of $O(n \cdot \log(n))$

With every iteration that I tried this, the run time of the first arrays were much higher than their subsequent ones. I'm not sure if that has to do with how the method is optimized in Java or if it is meant to be used with arrays of larger sizes.