Saman Kashanchi

I first set out to create all the different sorting algorithms in template classes. I used the class slides and information on zybook to help me create these functions. Simple algorithms such as Bubble sort and Insertion sort were rather easy to figure out. For Bubble sort I used a custom method called swap which is used to swap the values of two numbers if they met the required if condition. Both Selection and Bubble sort use nested for loops to iterate and compare the values in their Data set, this is why they are the two slowest methods for sorting with a big O of N squared. This also holds true for Insertion Sort as it also utilizes a nested loop inorder to compare and iterate through its values, however this method proves be slightly faster when dealing with slightly sorted data as it can sometimes by pass the second loop if values are already in order. The biggest challenge came with implementing the merge and quick sorting algorithms as they are more complicated and use recourgain in their functionality. When testing on a small data set of 100 rows, one could not recognize the difference in the time processing as there wasn't enough data to make a difference. When the data set grew to 1500 plus lines I started realizing a slight difference in the Epoch timestamp of the quick sort algorithm compared to others. I believe the main trade off of which algorithm to use mainly depends on the DataSet, the resources and the time ones given. Overall when dealing with a huge data set merge sorting is known to be the best and with smaller data set quick sorting. I found an interesting article online that said that one can even make sorting quicker by combining merge sort with quick sort. So the algorithm sorts the masses of data until it is much smaller and then switches to quick sort to fish the job off. The empirical approach of testing such algorithms is quite time consuming. Ones answer is also fitted to the specific data set they test on , if one is to do a proper empirical test one must have multiple testing data sets drawn from different sources which aain can be very time consuming and one can easily overlook a specific scenario.