Trenton Faillace

Assignment 6 Essay

In assignment six I learned about the vast differences between different sorting algorithms. At first glance when I ran the program and looked at the results of how long each algorithm took to sort a small data set, there was practically no difference as each time displayed as less than a second. Soon I realized that I should increase the number of elements that I would have the program sort up to one million. When the program was tasked with sorting 1 million elements, the time to compute the sorting was much slower. The first thing that stood out was that bubble sort took much longer than the rest of the algorithms to sort the data. Bubble sort took about 40 seconds to sort, insertion and selection both took about 10, and merge and quick both took less than a second. This was a vast difference that I was not necessarily expecting, I did not realize that merge and quick sort would be much faster than the rest of them, especially when dealing with larger data sets. I learned that the biggest trade off when selecting which algorithm to use came down to three things; how sorted the data you are working with is, complexity of implementation, and what resources you are dealing with. The implementation aspect just comes down to how much time the programmer is willing to put in to learn these algorithms and write them out. Quick sort and merge sort always seem to be the best choices for any relatively large data set if you have the resources to handle them. They might do things in a more complicated way, however they get the job done much faster than the alternatives. The method of using the c++ time library to analyze the run time of these algorithms is not a full proof method because it is not very precise, however for the purpose of this assignment it was good enough to display the clear differences between these algorithms and which ones are the best choices for differing scenarios.