Did the results of your program confirm what you already knew or suspected about these sorting algorithms? How?

The results of this program gave me some new insight into the way sorting algorithms work. I never had put much thought into how some algorithms, which utilizing more efficient methods, could actually decrease cpu time. I naively thought that modern computer processors were so strong that it wouldn't really matter. But, after working on this assignment, and running a huge amounts of monsters through these algorithms, and seeing that it can take literal seconds to sort through, really gave me a new insight into how efficiency in algorithms is crucial.

What differences did you see between and among the slow (bubble, selection, insertion) and fast (quick, merge, merge-insertion) algorithms?

Something I noticed that was different between the slow algorithms and the fast was that, among the slow algorithms, there is a much more significant amount of comparing going on. I think that this increase in comparing is one of the reasons why they are slower, because it has to constantly check many elements and doesn't go through the list utilizing more efficient methods like the splitting up in merge sort.