Lauren Szlosek - CPSC 350-03

Throughout the duration of the class a large portion of what we have learned has been extremely conceptual. Meaning, most of the time we discuss the theory or we look at the pseudo code. However, I am a visual learner and big conceptual theories have a hard time sticking for me. While taking Java it took me weeks to understand classes until I implemented them in my Visual Programming class. In regards to this assignment, we have been learning about time and space complexity. This made sense in a very basic way but I was still confused how different runtimes could have a significant impact from algorithm to algorithm. Merge sort has a runtime of O(n logn) because it uses recursion, and uses another linear function. Selection sort has a runtime of O(n^2) because of the nested for loop. Shell sort has a time complexity of O(n^2) because of the nested for loop as well. Due to just these three facts it should conclude that merge and shell sorts should have shorter runtimes than the selection sort. This is true, as when my programs were run with the 10k file merge took 10375 microseconds, shell took 1817 microseconds and selection sort took an entire 923434 microseconds making it last place by a landslide. While it should be that shell and merge have similar runtimes, they aren't exactly the same but they are much closer together than the selection sort algorithm is. Coming back to what I mentioned in the beginning of this paper, seeing the three algorithms definitely helped me understand how different big Os could change the runtimes of programs.