

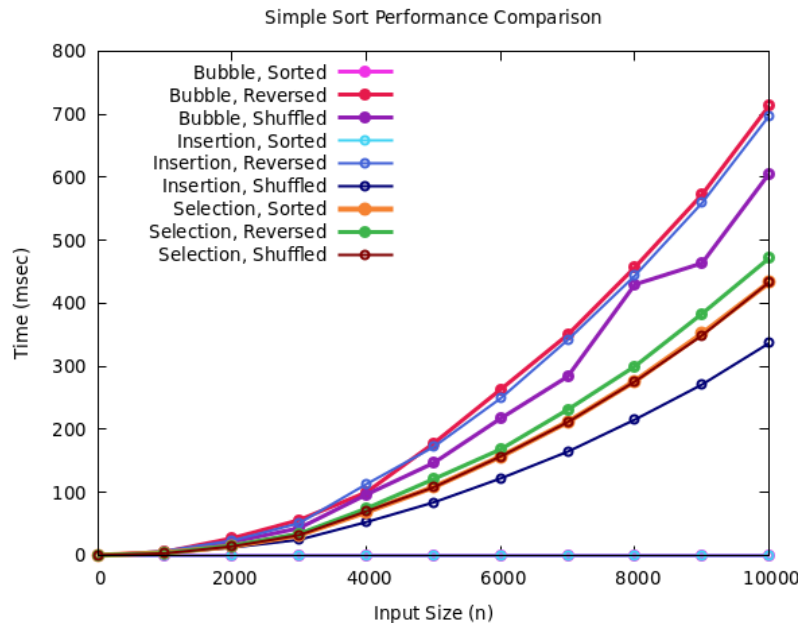
NAME: Ben Puryear

FILE: HW-1_WriteUp.pdf

DATE: Fall 2021

HW-1 Write Up

Graph:



Explanation:

According to the graph the slowest scenario is sorting a reversed array using bubble sort. This makes sense because bubble sort's downside is that it has the potential to have to do a lot of swaps and loops, resulting in a longer time. On the opposite side of the spectrum is sorting a sorted array using bubble sort or insertion sort. These two were the fastest methods by far when in the right conditions because they had no swaps, and only traversed the array once.

Problems:

My original push included an incorrectly written insertion sort that forgot to stop after it had found the correct position and kept on traversing the array. This led me to write another few sentences. "One thing to notice is that all variations of selection sort are extremely close to each other. I believe this happens because of how selection sort works by having a set number of traverses through loops, meaning that the only time that can be added is if it needs to swap values."