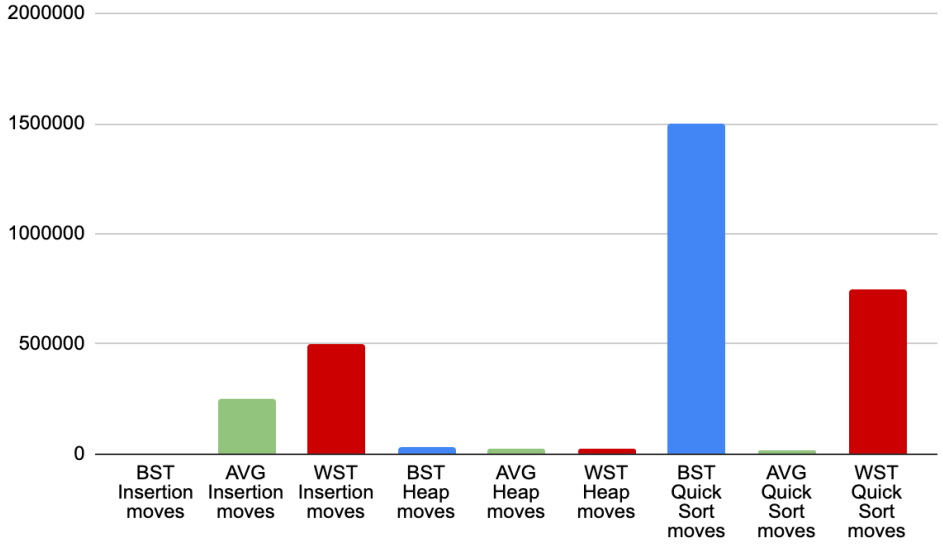
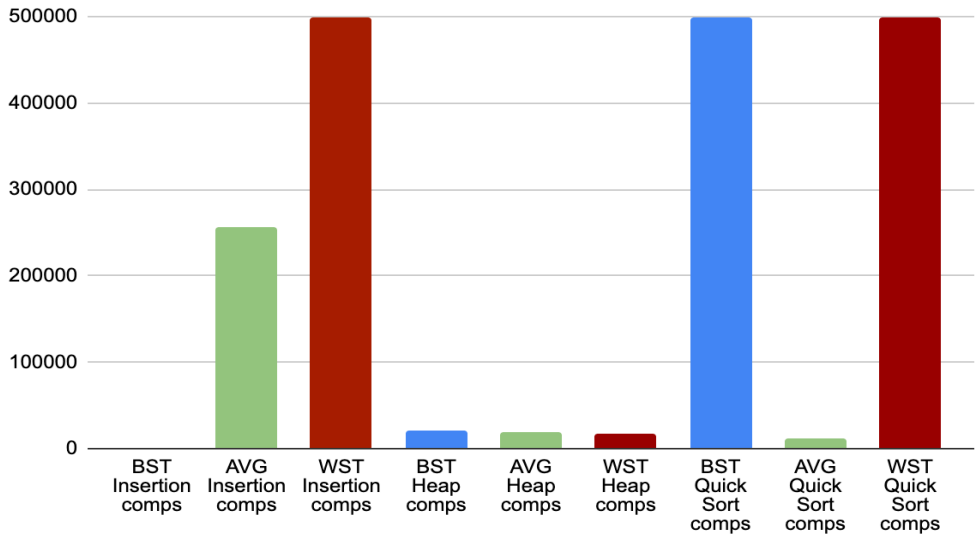
Homework04

Hwijong Im

Problem 01

Implemented 3 Sorting Algorithms(Insertion, Hash Table, Quick). For each sort algorithm, I marked moves and comps to record the number of moves and compare elements in the data structure. The number of moves and comps of best, average, and worst cases will be written in the txt file, which represents 9 results. And then, I made a simple Check function so that the sort algorithms ran properly and sorted the arrays correctly. Lastly, down below, there are plot of results that I implemented with Python. Each plot represents the number of moves and compares.





1. The insertion sort algorithm results in zero moves when the best-case scenario because only ‘If’ condition is untrue, the algorithm does not loop. Therefore, if values are sorted properly, the algorithm will not initiate the loop.

2. The number of iterations showed 999 because a comparison must be done once each iteration to show that the element does not need to be moved. Therefore, the comparison will continue to the next element as the key.

3. Based on two plots and metrics, Heap Sort is the most consistent in performance. However, the Insertion sort gets the highest performance in the best case, and Quick sort showed the best performance on average, but the worst moves and comps on BST and WST cases. Consequently, Heap sort is the most consistent as the moves and comparisons in all ways.