**CS/CPE590: Algorithms**

**Spring 2023**

Assignment1

Chao Zheng

Lab Report

02/23/2023

**Result:**



Table1: Merge Sort Runtime in ms (Average Summary)

Figure 1: Merge Sort Input Size vs Runtime (n = 10)

Figure 2: Merge Sort Input Size vs Runtime (n = 25)

Figure 2: Merge Sort Input Size vs Runtime (n = 50)



Table2: Insertion Sort Runtime in ms (Average Summary)

**Discussion:**

After implementing both sorts with visualized data and graphs, they both have the advantages to process some specific data and shortcomings. In general, the most common comparison is the big O of both sorts, merge sort is overwhelmed either analyze from the average complexity or worse case complexity aspects, which we can verify from the data table above that insertion sort cannot process a tremendous amount of data efficiently, it processes relatively slow even with some acceptable size of data. From Big O nation, we can analyze the worst case of O(n^2) and the average case is close to the worst case, as a result of random vector and inverse sorted vector consuming most of the time while running the project. For sorted vector result of insertion sort is much faster than merge sort, since it just compares the lengths of two rows by the ivertor\_lenth function and does not run the while loop, which leads the result close to O(n). Besides the runtime comparison, the insertion sort would reach the runtime error and terminate the program with the amount of data and dimension increase as the “- “appears in Table 2.

Merge sort is considered a balanced performance sort, no matter to proceed what type of data and regardless of the amount of data and dimension of vectors, it gives acceptable runtimes during the experiment. Related to the Big O of merge sort O(nlog(n)) and the graph above, the runtime increases slowly when the amount of data is under 1000000, the runtime increases nearly linear once it exceeds the threshold, therefore, we can assume there exist a more efficient sort to process the extreme amount of data, since it requires to separate them into subarray and compare the value.