**Report**

**Goal:**

The goal of this assignment is to conduct a comparative benchmark between the six sorting algorithms; Selection, Bubble, Insertion, Heap, Merge and Quick sort. By running the six algorithms against integer data collections with different sizes and different presorted order and measure the running time of each algorithm. Furthermore, plotting the running time for each sort algorithm and verify the rate of growth of each of them.

**Assumptions:**

The assumption is to compare the theoretical results with the practical ones. First, the theoretical values for the big O for selection is O(n^2), for Bubble is O (n^2), Insertion is O (n^2), Heap is O (n log n), Merge is O (n log n), Quick is O(n^2). By analyzing the relation between the input and the time variation will confirm the assumption which has been made.

**Design and Analysis:**

**Random data selection sort:**

after obtaining the data from each of the sorting algorithms we draw graph between the data random input and time of each algorithm in micro seconds. The graph indicates that for the random data Quick, insertion and selection have greater complexity than Bubble, Heap and merge. However, theoretical Quick, bubble, insertion and selection have complexity of O(n^2). While merge and heap have complexity of O(n log n ).

**Presorted Dataset in Ascending Order:**

for the presorted Dataset from all the algorithms significantly decreased and hence the run time of them decreased; however, selection sort does not change and still have the same big O notation

**Presorted Dataset in Descending Order:**

the same thing happens in the presorted data in descending order and all of them decreases but here the bubble sort remains un changeable and did not get affected with the presorting.