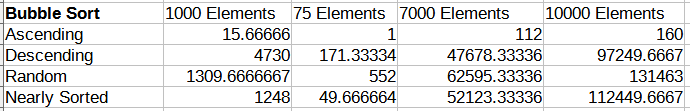
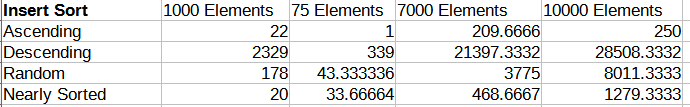
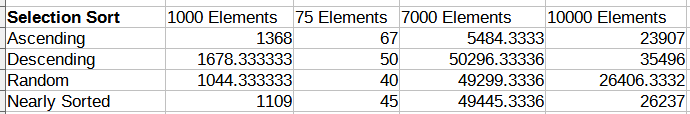
**Time Complexity Analysis – Sorting Algorithms**

Rebecca Massey 20540609

The bubble will always make a full pass at least once to compare all data in the array and in the best case will not make any swaps (O(N) complexity). However if not the best case then the bubble sort will perform N- P comparisons every pass while swapping. This data shows that the worst case scenario on average is the descending order and random data sets which take much longer than the rest even when the sample size is small due to the algorithm performs many swaps and comparisons.

In comparison to the bubble sort, selection sort has more balanced time complexity when sorting ascending, descending, random and nearly sorted data sets. Selection sort does N – 1 passes total where N is the number of elements, only swaps the smallest element, and conducts a decreasing number of comparisons per pass. This is reflected approximately in the tabulated results with all ordered cases having close to the same execution speed.

Insertion sort has the same best case as the bubble sort which occurs when no elements must be moved (O(N)) similarly if the elements are reverse ordered then the algorithm must compare all elements in every pass. As reflected in the table the ascending order is the best case and the descending order is the worst case, taking far longer than all others to sort. Unlike the selection sort the insertion sort is capable of taking advantage of almost sorted data resulting in better times on average for random and nearly sorted data. This is due to the insertion sort directly placing the data into the perceived correct position instead of swapping the elements constantly however it still performs a lot of comparisons.