**CSCI B505 – Programming Assignment 1 -** Bubble sort vs Insertion Sort

**1. Plots**

**1.1 Plot 1 - Random Inputs**

For Random input, the execution of insertion sort is efficient than Bubble sort

**1.2 Plot 2 - Non decreasing inputs**

For non decreasing input, Insertion sort out performed Bubble sort and the performance is nearly linear

**1.3 Plot 3 - Non increasing inputs**

For the non-increasing inputs, bubble sort out performed insertion sort

**2. Functional details**

**2.1 Platform**

All the metrics are calculated in a single machine with below configuration

Processor: 2.5 Ghz Intel core i7

Memory: 16 GB

Operating System: macOS Sierra

Tools: Pycharm, Microsoft Excel

**2.2 Coding/plotting choices**

The coding is done in Python 2.7 and plotting is made with Microsoft office tools.

**2.3 Data files**

Data used for sorting is generated with a Python script which generate 3 different types of files with integer elements ranging from 2000 to 50000 on each.

1. Random elements (R1.txt,R2.txt….R25.txt)
2. Non Decreasing elements (I1.txt,I2.txt….I25.txt)
3. Non Increasing elements (D1.txt,D2.txt….D25.txt)

All these files are persisted in the disk and sorted in memory.

**2.4 Observation**

a. The completion time is high for Python  
b. Program consumed 98% of the processor and it executed in a single thread  
c. Usage of “xrange” instead of “range” retrieved better results

**3. Conclusion**

Both the algorithm displayed quadratic running time for Random input with Insertion sort performing better than bubble almost by 8 times on the same data set and the execution time of bubble sort hiked with increase in data which is evident after the record count crosses 23 thousand.

In the second instance(plot 2), the data is already sorted, hence, insertion sort’s execution time is nearly logarithmic, however, Bubble sort is quadratic as it has to iterate twice on all the elements to complete the execution and time consumption is similar as plot1.

In the third instance(plot 3), we can consider the data as worst case scenario for both algorithms, here Bubble sort out performed Insertion as only 2 iterations are required for each element to complete the execution whereas in insertion sort multiple iterations are involved.

The overall observation shows that the Insertion sort performs better in terms of data volume and at best case/average case scenarios, hence, we can use Insertion sort in real life where sorting as to be fast and the input size is huge, whereas, Bubble sort performs better with any real life scenario provided the data volume is low.