**Kathmandu University**

**Department of Computer Science and Engineering**

**Dhulikhel, Kavre**

****

**LAB 1**

**[Code No: COMP 314]**

**Submitted by**

**Paribartan Timalsina**

**Roll no:56**

**Group: Computer Engineering**

**Level: III year/II sem**

**Submitted to**

**Dr.Rajani Chulyadyo**

**Department of Computer Science and Engineering**

**Introduction**

In Lab 1 we wrote different algorithms and tested the validity of those algorithms through different test cases.At first a simple function was made that sums the elements present in array and return the sum.This function was tested using the unittest library.The test cases were of different types where the number of elements and the value of elements passed  were different and all of these cases were ran.These test validated if our algorithm was correct or not.After that we wrote the algorithm for insertion sort and selection sort and then we wrote the test cases to validate these algorithms.For validating the sorting algorithms we used the array as input and returned sorted array as output.In the test cases we passed array of different sizes and both positive and negative numbers were used for testing.These tests were run and we verified our algorithm was correct.

After this we plotted the graph to find out how the resource consumption is increased when we increase the size of the array.The graphs were plotted for insertion sort having the general,best and worst case and selection sort.

**Outputs:**

The output of running the test cases for sum of numbers in array is:

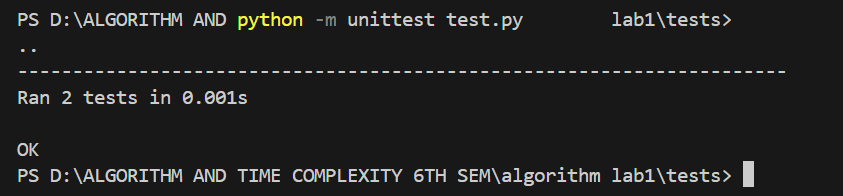


Figure: Output of testing sum function

The output of running test cases for sorting algorithms are:

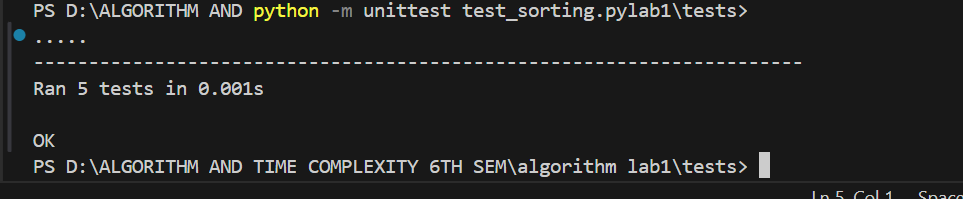


Figure: Output of testing sorting algorithms

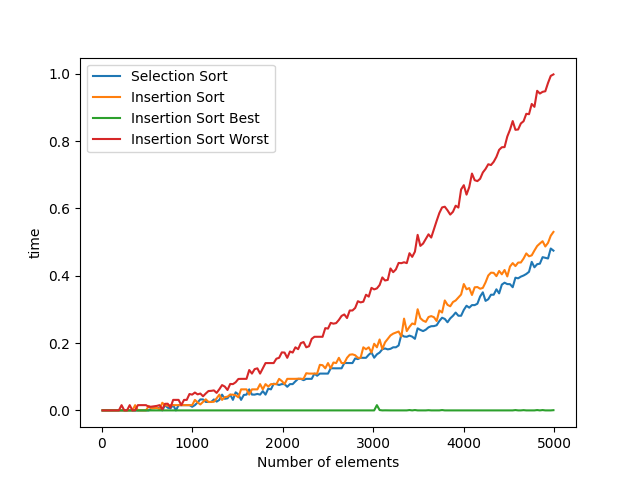
After these test cases the graph was plotted and the graph was obtained as:

Figure:Graph of comparision between different algorithms

**Observation:**

From the above graph we can see that the plot of time taken by sorting algorithms as the number of elements to sort increases is given.The blue line shows the selection sort curve.The curve shows an increase in sorting time as number of elements increases.The theoretical time complexity of selection sort is O(n2) and the curve plotted has exactly shown the quadratic nature.

The green line gives the curve of insertion sort in best case.This condition occurs when the input array is already sorted.In this time the time complexity is O(n) as it doesn’t need any swapping.The flat line at the bottom shows the nature indicating that time taken isn’t significantly increased as number of elements grows.

The orange line gives the insertion sort in the general case.It has the time complexity of O(n2).This curve is similar to the selection sort and both perform in the same way in the general case.

The red line indicates the worst case scenario of insertion sort.This occurs when the array to be sorted is in reverse order and hence need for maximum number of swaps.

From all these we can see that selection sort and general case of insertion sort appears same and are slightly better than the worst case.Insertion sort best case outperforms all the cases significantly showing nearly constant time complexity.The worst case of insertion sort increases more steeply than general case indicating more pronounced O(n2) behaviour.The observations that we obtained might not be In accordance to the theoretical conditions because our size if elements are very small and there are hardware and performance limitations in our condition that can’t be tested.

The code for this lab can be found at :

<https://github.com/Paribartan-Timalsina/COMP314LAB1TESTCASES.git>