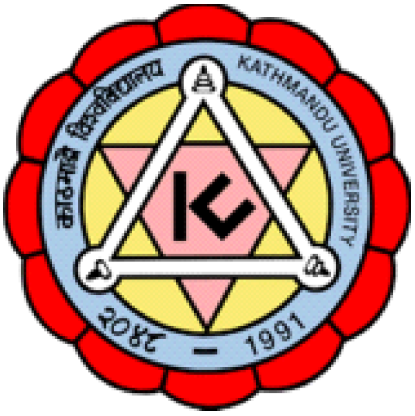
KATHMANDU UNIVERSITY

DHULIKHEL, NEPAL



COMP 314: Lab 1 Report

SUBMITTED BY

Bishal Neupane

Roll No: 31

Group: Computer Engineering

Year/Semester: III/II

SUBMITTED TO

Rajani Chulyadyo, PhD

Assistant Professor

Department of Computer Science and Engineering

DESCRIPTION:

In this lab, we learnt about validating the working of algorithms and programs using test cases. Initially, a simple function sum was tested using unittest library provided by Python. The sum function adds the numbers present in the passed array and returns the final sum. The test cases were to test for valid sum, invalid sum, different sizes array and empty array.

The testing knowledge learnt in this simple exercise was utilized to validate the correctness of written sorting algorithms namely insertion sort and selection sort. The test cases used were to test for working of algorithms with empty lists, lists of a few different sizes, lists containing negative numbers, positive numbers and lists containing sorted numbers.

Finally, a graph was made to compare the time complexity of the selection sort and insertion sort(best, general and worst cases).

Output:

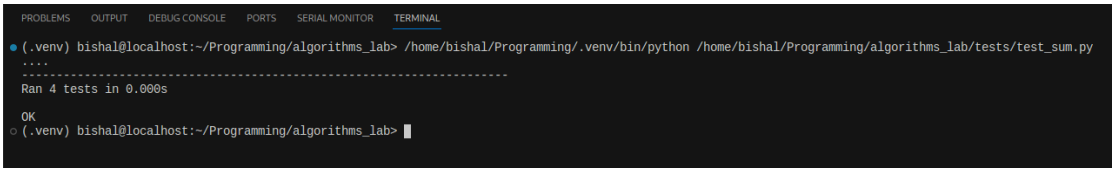


Image 1: Testing sum function using various test cases utilizing unittest library

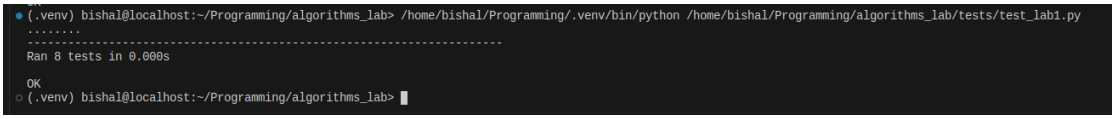


Image 2: Testing insertion sort and selection sort using various test cases

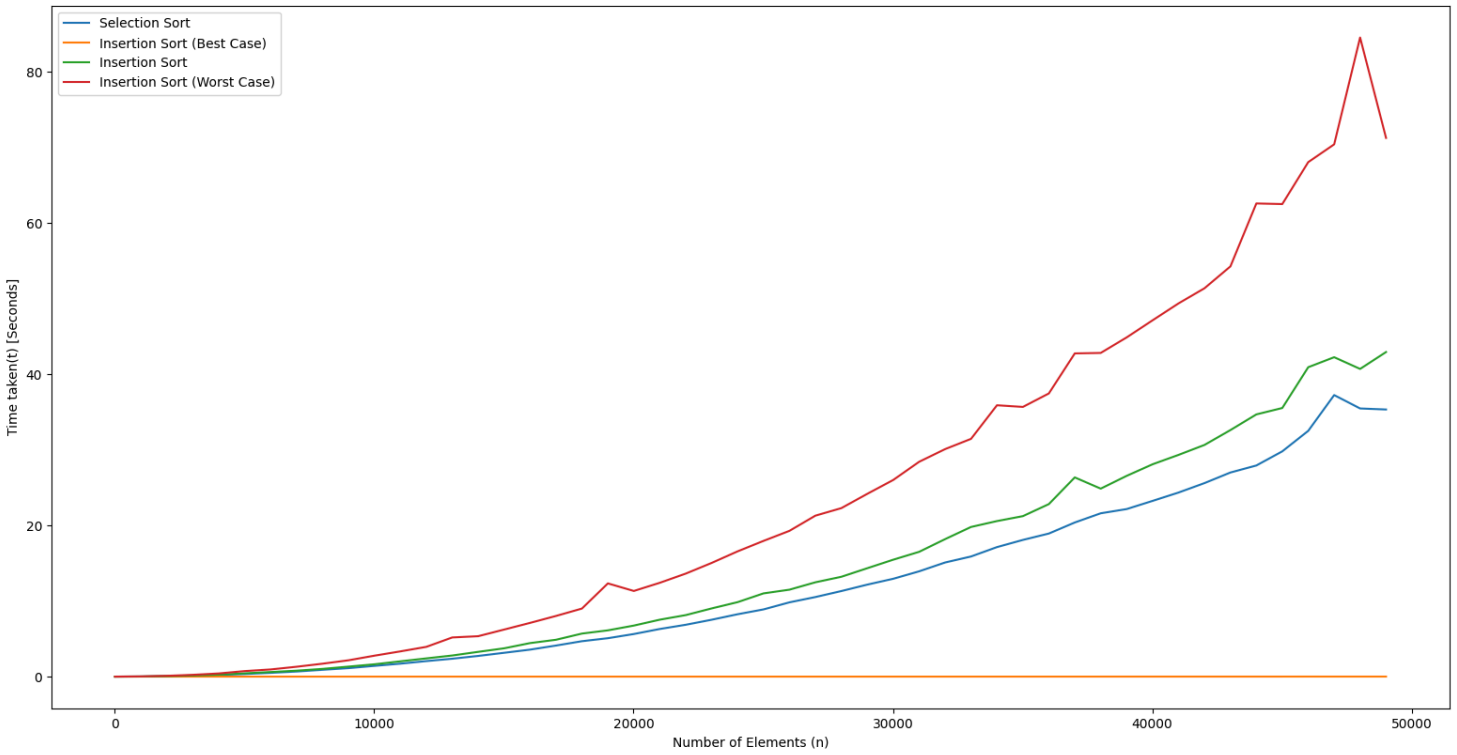


Image 3: Output of comparison between time taken for Selection Sort, and Insertion Sort (General, Worst and Best cases)

Observation:

The theoretical time complexities are:

- Selection Sort(all cases): $O(n^2)$
- Insertion Sort(worst case): $O(n^2)$
- Insertion sort(Best case): $O(n)$

These theoretical complexities will only hold after a certain $n > n_0$. From the graph, it is clear that insertion sort(best case) performed the best while insertion sort(worst case) performed the worst, selection sort performed better than insertion sort(general). The observation not being per the theoretical time complexities might be because of the smaller list size. However, because of hardware and computation limitations, an even higher number of data couldn't be tested.

The code can be found at: https://github.com/earthPerson-001/algorithms-lab/