DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Faculty of Engineering
University of Moratuwa

CS 2022: Data Structures & Algorithms

Lab 1: Performance of Insertion Sort

Objectives: The aim of this lab session is to provide you with the understanding of implementing insertion sort and the performance of insertion sort.

Learning Outcomes: After completing this lab the students should be able to;

1. Explain the insertion sort.
2. Explain the time complexity of insertion sort.

In this lab session you will;

- Implement insertion sort in python.
- Test the correctness of implementation using appropriate sample inputs.
- Evaluate the time taken by insertion sort for inputs of different sizes.

You are expected to use python programming language for completion of this lab exercise. You should upload your completed code in separate .py files and a report (in pdf format) explaining how the running time of your program increases with the increase in the input size to the assignment in the Moodle page.

Task 1: *Implementing the Insertion Sort*

In the first task, you are expected implement the insertion sort algorithm discussed in the class. The signature of the insertion method should be;

```
def InsertionSort(NumberList)
```

Include your code in a file named **is_<your-index-number>.py (for example is_150150X.py / is_140140.py)**.

The pseudo code of the insertion sort is shown below (we are assuming the array/list index starts from 0).

```
INSERTION-SORT(n, A)
1. for j = 1 to n-1
2.   key = A[j]
3.   //Insert A[j] into the sorted sequence A[0, 1, ……, j-1].
4.   i = j-1
5.   while i >= 0 and A[i] > key
6.        A[i+1] = A[i]
7.        i = i-1
8.   A[i+1] = key
```

Task 2: ***Testing the Implementation***

As the second task, you are expected to write code to test the correctness of the method (insertion sort) you implemented. You must test your code with **at least four (4) different inputs**. **Please make sure that your test cases cover all the possible scenarios**. The signature of the testing method should be;

```
def TestInsertionSort(NumberList)
```

Include your testing code in a file named **istest_<your-index-number>.py (for example istest_150150X.py / istest_140140.py)**.

Task 3: ***Performance of the Insertion Sort Implementation***

As the third task, you are expected to test the time taken for your insertion sort implementation to sort arrays of different sizes. You should generate random number lists of following sizes and check the time taken by your code to sort the random arrays. The sizes of the arrays are 100, 500, 1000, 2000, 3000, 4000, 5000, 7500, 10000. For measuring the time taken for execution you can import the " time" module (using "`import time`") and "time.time()" function. For generating random numbers, you can import "random" module (using "`import random`") and use "`random.randrange(min_number,max_number+1, step_size)`" function. Sample code on how to use this is included in Moodle page. You are expected to upload a report with your observations in a tabular format and graph format as the submission for this task.

Name your report as **t3_<your-index-number>.pdf (for example t3_150150X.pdf / t3_140140.pdf)**.