Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

Algorithm and Complexities Lab Report 01

on

‘**Insertion & Selection Sort - Time Complexities**’

Submitted By:

Reewaj Khanal (61)

Submitted to:

Dr. Rajani Chulyadyo

Assistant Professor

Department of Computer Science and Engineering

School of Engineering

Kathmandu University

Dhulikhel, Kavre

Submission Date: Thursday 2 May 2024

1. Purpose

Implementation, testing and performance measurement of sorting algorithms.

2. Tasks

2.1. Implementing the following sorting algorithms [Code in Python]:

(a) Insertion sort

```python
# Insertion Sort

def insertion_sort(arr):

   for i in range(1, len(arr)):

       key = arr[i]

       j = i - 1

       while j >= 0 and key < arr[j]:

           arr[j + 1] = arr[j]

           j -= 1

       arr[j + 1] = key
```

(b) Selection sort

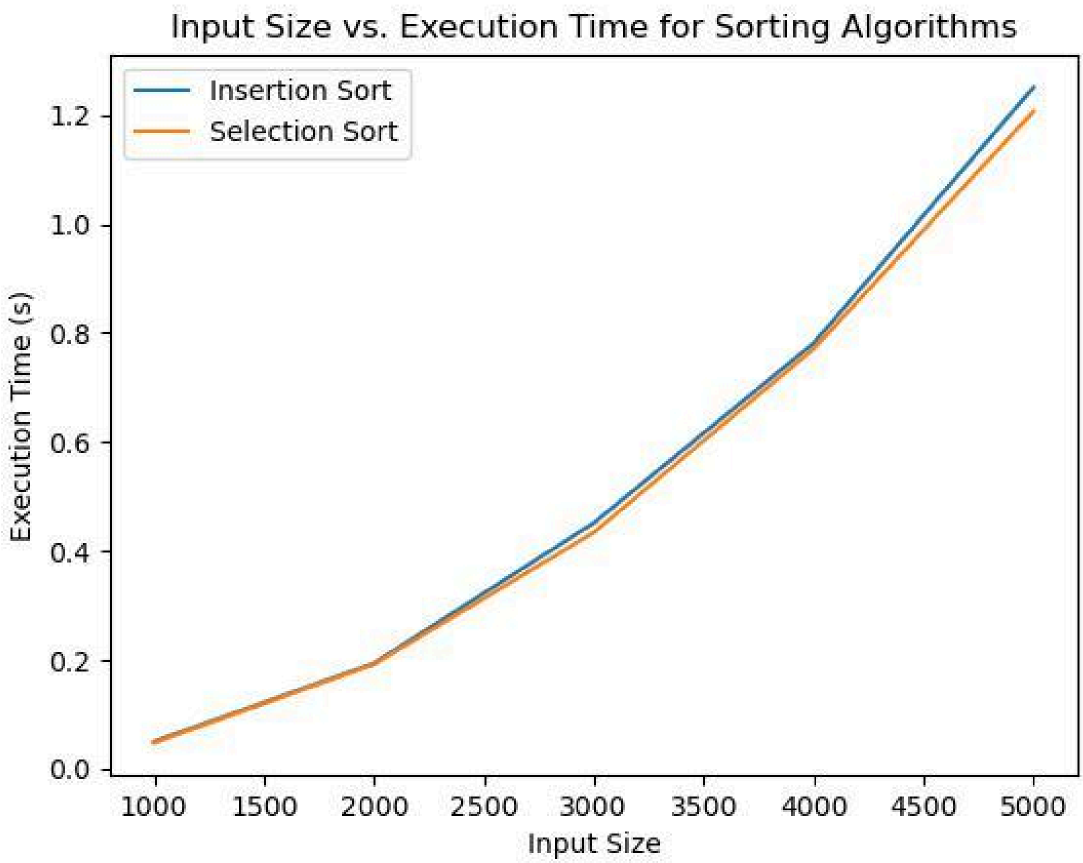
```python
# Selection Sort
def selection_sort(arr):
   for i in range(len(arr)):
       min_idx = i
       for j in range(i + 1, len(arr)):
           if arr[j] < arr[min_idx]:
               min_idx = j
       arr[i], arr[min_idx] = arr[min_idx], arr[i]
```

2.2. Some test cases to test my program.

```python
# Test cases for sorting algorithms
def test_sorting_algorithms():
   # Test case 1
   arr1 = [3, 2, 1, 4, 5]
   insertion_sort(arr1)
   assert arr1 == [1, 2, 3, 4, 5]

   # Test case 2
   arr2 = [5, 4, 3, 2, 1]
   selection_sort(arr2)
   assert arr2 == [1, 2, 3, 4, 5]
```

3. **Some random inputs for my program and apply both insertion sort and merge sort algorithms to sort the generated sequence of data. Recorded execution times of both algorithms for inputs of different sizes. Plotting an input-size vs execution-time graph.**



4. **My observations.**

The plotted graph shows the execution time of Insertion Sort and Selection Sort for sorting algorithms with varying input sizes. The x-axis represents the input size and the y-axis represents the execution time in seconds.

The graph suggests that Insertion Sort has a better time complexity than Selection Sort. As the size of the input increases, the execution time for Selection Sort increases at a much faster rate than Insertion Sort.

This is because Insertion Sort is an O(n^2) algorithm, which means that the execution time grows quadratically with the input size. Selection Sort, on the other hand, is also an O(n^2) algorithm, but with a higher coefficient, resulting in a steeper slope on the graph.