**Course No :** CSE 204

**Course Name:** Data Structure and Algorithm Sessional

**Offline No:**  2

**Problem No:** 1

**Problem Name:**

Comparison of Selection Sort and Insertion Sort

**Submitted By:**

Name: Kawshik Kumar Paul

Student ID : 1705043

Section : A2

Level : 2, Term : 1

BUET CSE’17

**Time Complexity Analysis :**

**Selection Sort :**

Selection sort has O(n²) time complexity. O(n²) isn’t a good time complexity for sorting lists

when it comes to large input sizes. This algorithm sorts an array or list by repeatedly finding the

minimum value (if we are sorting in ascending order) from the list or array and placing it at the

beginning of the list.

To sum up the running times for selection sort:

• Worst case: Θ (n^2)

• Best case: Θ (n^2)

• Average case for a random array: Θ (n^2)

**Insertion Sort:**

Insertion sort is based on the idea that one element from the input elements is consumed in each iteration to find its correct position i.e, the position to which it belongs in a sorted array.It iterates the input elements by growing the sorted array at each iteration. It compares the current element with the largest value in the sorted array. If the current element is greater, then it leaves the element in its place and moves on to the next element else it finds its correct position in the sorted array and moves it to that position. This is done by shifting all the elements, which are larger than the current element, in the sorted array to one position ahead. Insertion sort takes maximum time to sort if elements are sorted in reverse order. And it takes minimum time (Order of n) when elements are already sorted.

To sum up the running times for insertion sort:

• Worst case: Θ (n^2)

• Best case: Θ (n)

• Average case for a random array: Θ (n^2)

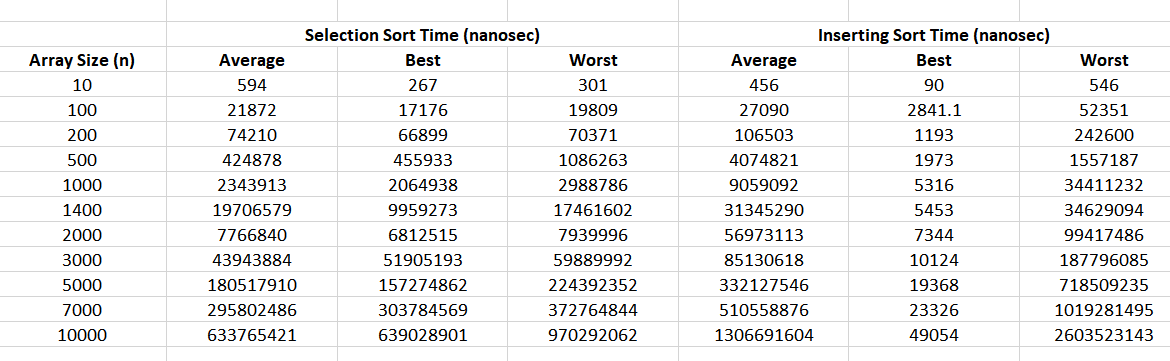
**Machine Configuration:**

Processor : Intel Core i7-8550U CPU @1.80GHz 1.99GHz

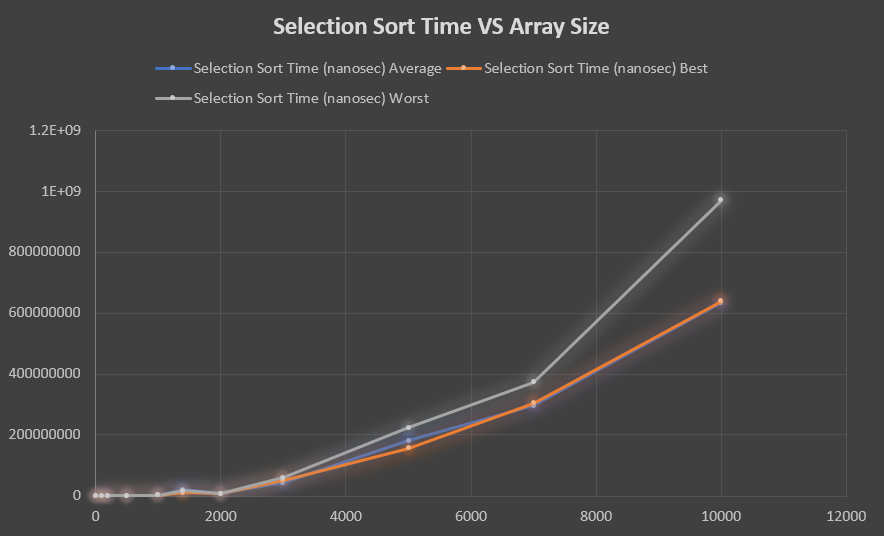
RAM : 8.00 GB

System Type : Windows 10 (Home) 64Bit

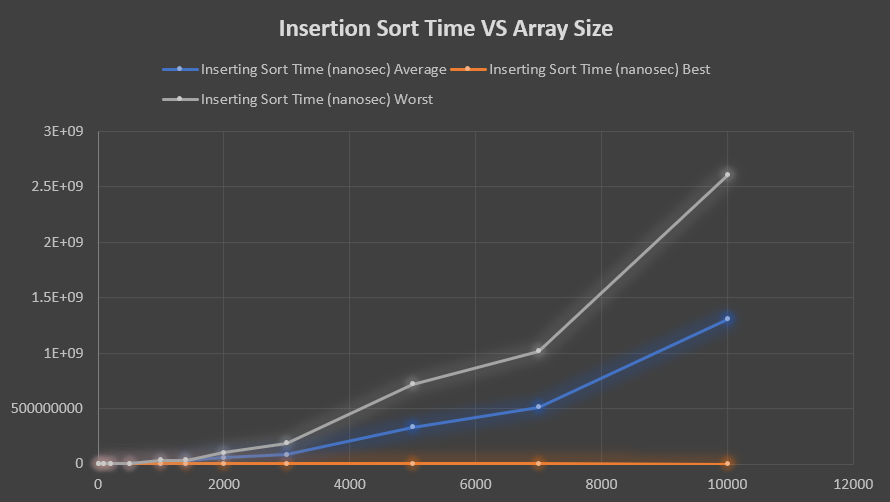
**Data Table for determining Best Case, Average Case, Worst Case:**



**Graph of Selection Sort:**



**Graph of Insertion Sort:**



**Graph of Comparison of Selection and Insertion Sort:**

