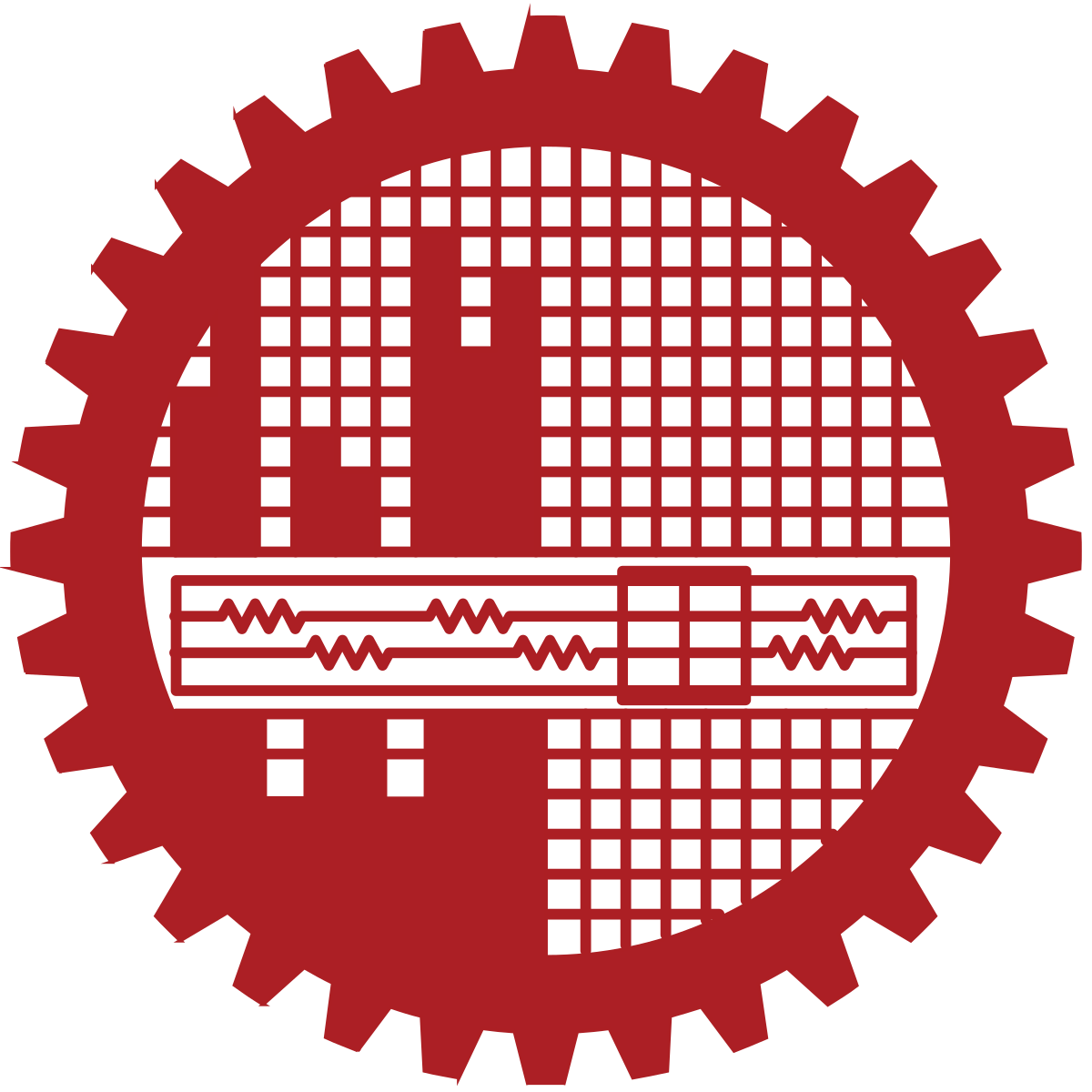
**CSE 204**

**DATA STRUCTURES AND ALGORITHM**

**SESSIONALS**

****

**OFFLINE NO : TWO (Problem 1)**

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

**SUBMITTED BY :**

**FAHMID - AL-RIFAT**

**STUDENT NO : 1705080**

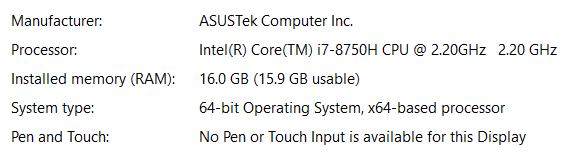
LEVEL-2 ,TERM-1

DEPARTMENT OF CSE

BUET

**OBJECTIVE:** The objective of offline is to compare the time needed to execute the two famous algorithms of sorting insertion sort and selection sort where various size of array are randomly generated for each time of execution where insertion sort data are sorted taking elements one by one where in selection sort data are sorted taking element and sorting them in according position

**MACHINE CONFIGURATION:**

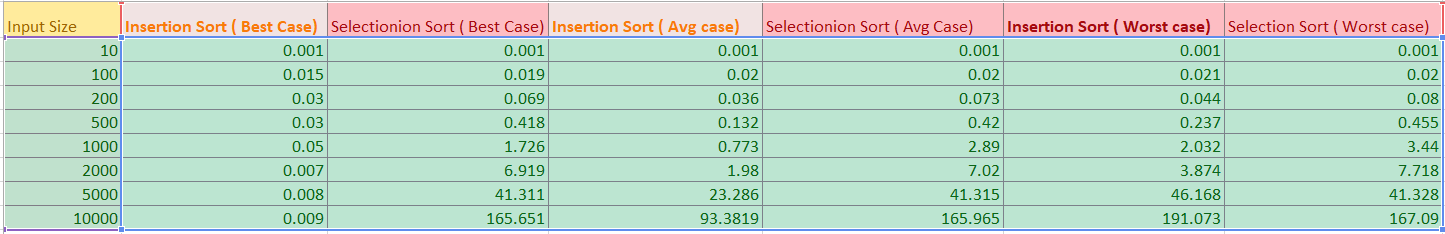


Compiler : Online Compiler GCC 8.1.1 ( jdoodle)

Code::Blocks 17.12

**DATA SET :**

**Table of sort algorithm time in millisecond and array size:**



**DISCUSSION :**

Insertion sort and selection sort are algorithm for sorting elements in the list of finites elements in array . In the code arrays elements are generated randomly . In the insertion sort each elements of the array taken serially and compare them with their before elements and place them in their proper position . Thus the array is sorted and complexity of the algorithm become **O(n2)** . In the best case the algorithm complexity becomes **O(n)** and the worst case **O(n2)** . On the other hand each element of the array taken serialy and swapped them with the minimum one .

Thus, running time complexity of selection sort is **O(n2)**.

= (n-1)+(n-2)+………..+2+1

= n(n-1)/2 = O(n2)

But in the best case complexity here do not decreased a lot , the complexity remain **O(n2)** as in this case we also have to traverse whole .

Among both of the sorting algorithms, the insertion sort is fast, efficient, stable while selection sort only works efficiently when the small set of elements is involved or the list is partially previously sorted.

I

