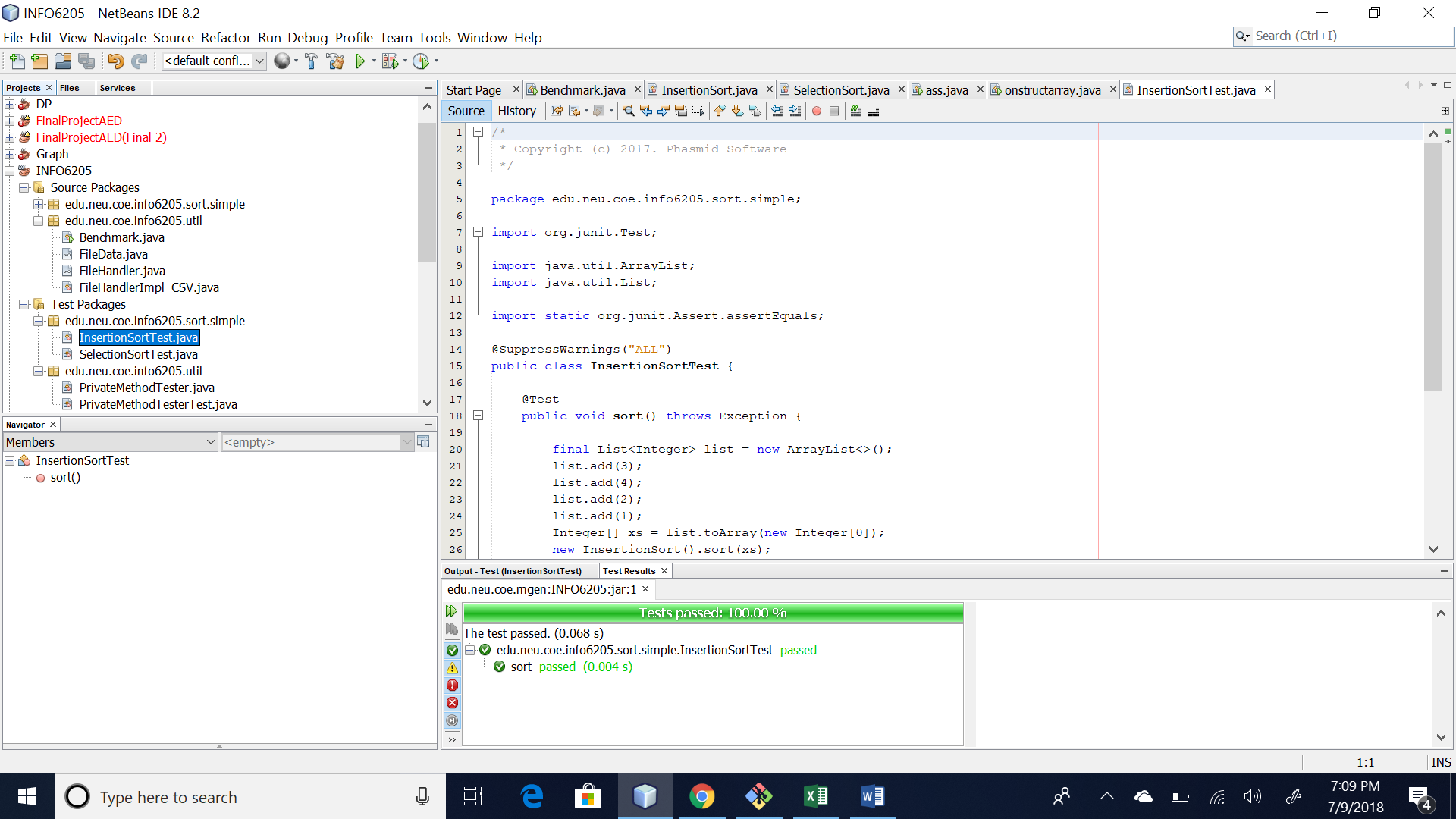
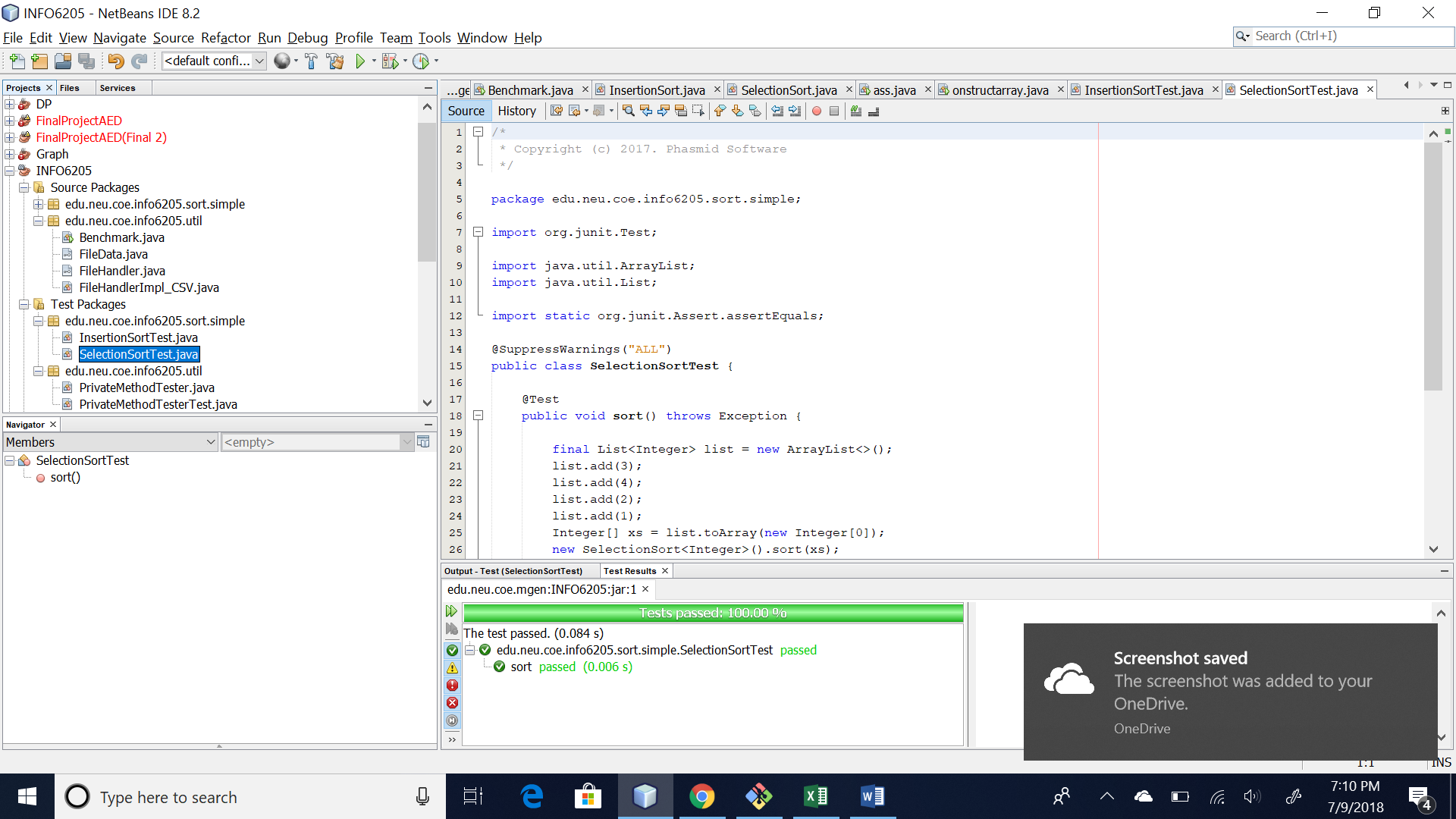
**Assignment 3**

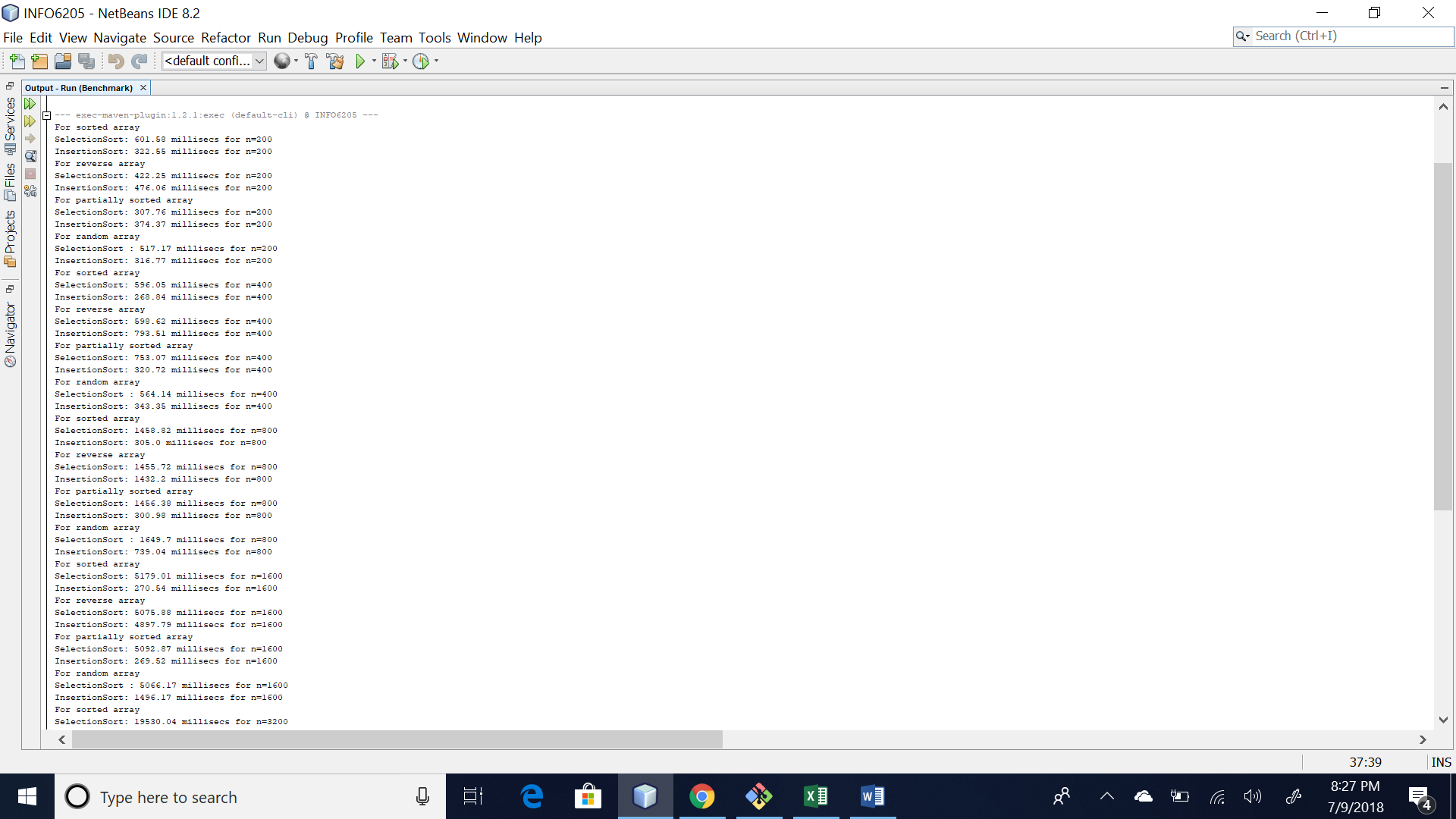
Passed test case for Insertion sort.



Passed case for Selection sort.



Running test for benchmark:



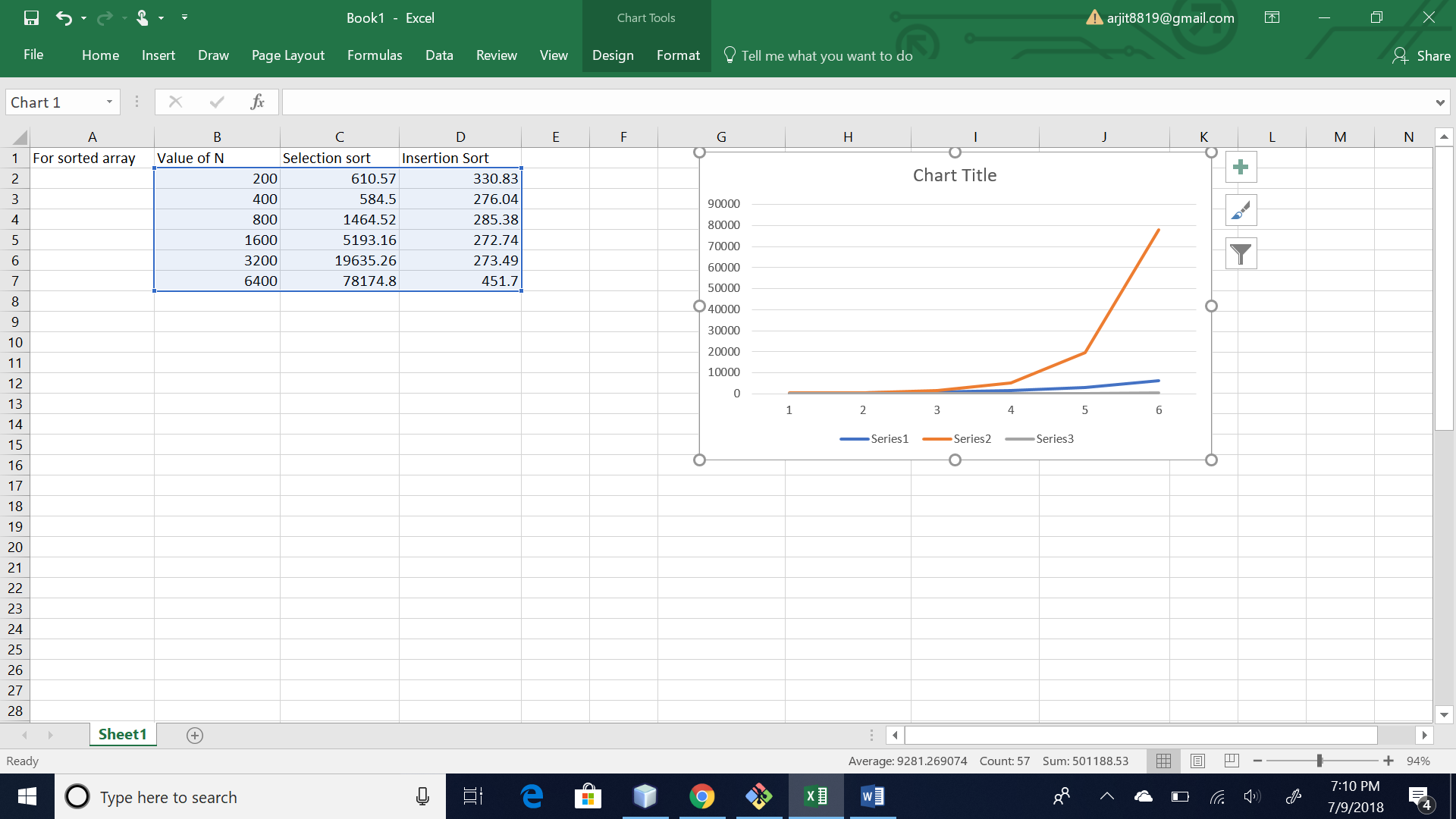
Below is the Reading for sorted array:

When the arrays are sorted Insertion sort takes only n-1 compares and 0 exchange where as the selection sort takes N^2/2 compares and N exchange which increases its time complexity drastically.

This can be seen through readings as well for N=200 . Insertion sort takes 300.83 milliseconds

Whereas Selections sort takes 610.57 seconds.

Similarly for N = 400 Insertions sort takes 276.04, Slection sort = 584.5.



Below is the Reading for Reversed array:

For reverse sorted arrays the time complexity for both the algorithms will be same because the number of compares for both are same like N^2/2.

This could also be seen in the readings like :

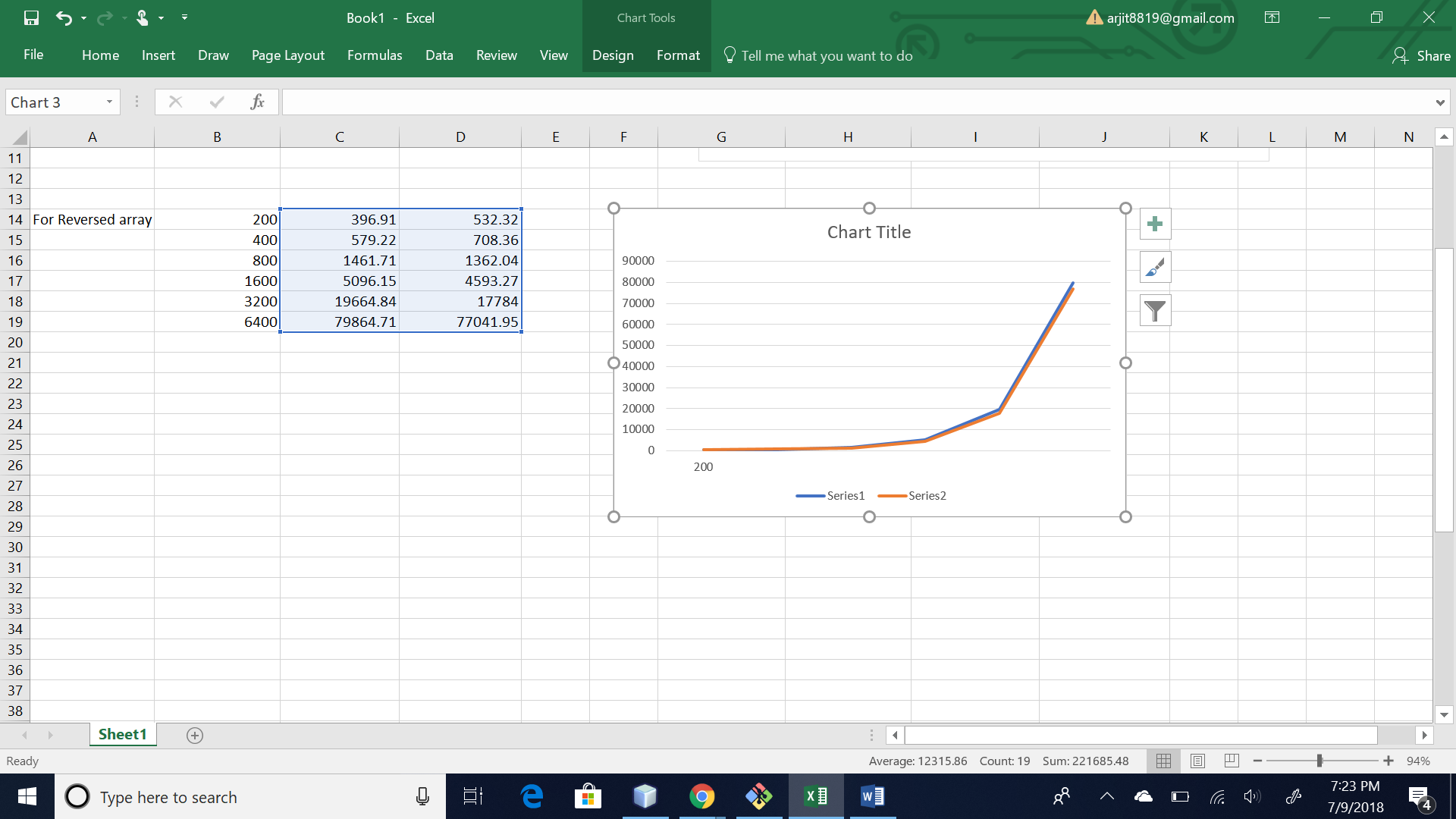
For value of N =200:

Insertion sort average time is 532.23

Selection sort is 396.91.

Which are close values

for



Below is the Reading for Partially sorted array:

For patially sorted array insertion sort runs in linear time and number of exchange equal to number of inversion. So for partially sorted again number of compare for insertion sort is less than the number of compares for selection sort.

So again the time complexity of selection sort is greater than that of insertion sort.

This could also be observed form the readings.

For example for N=400.

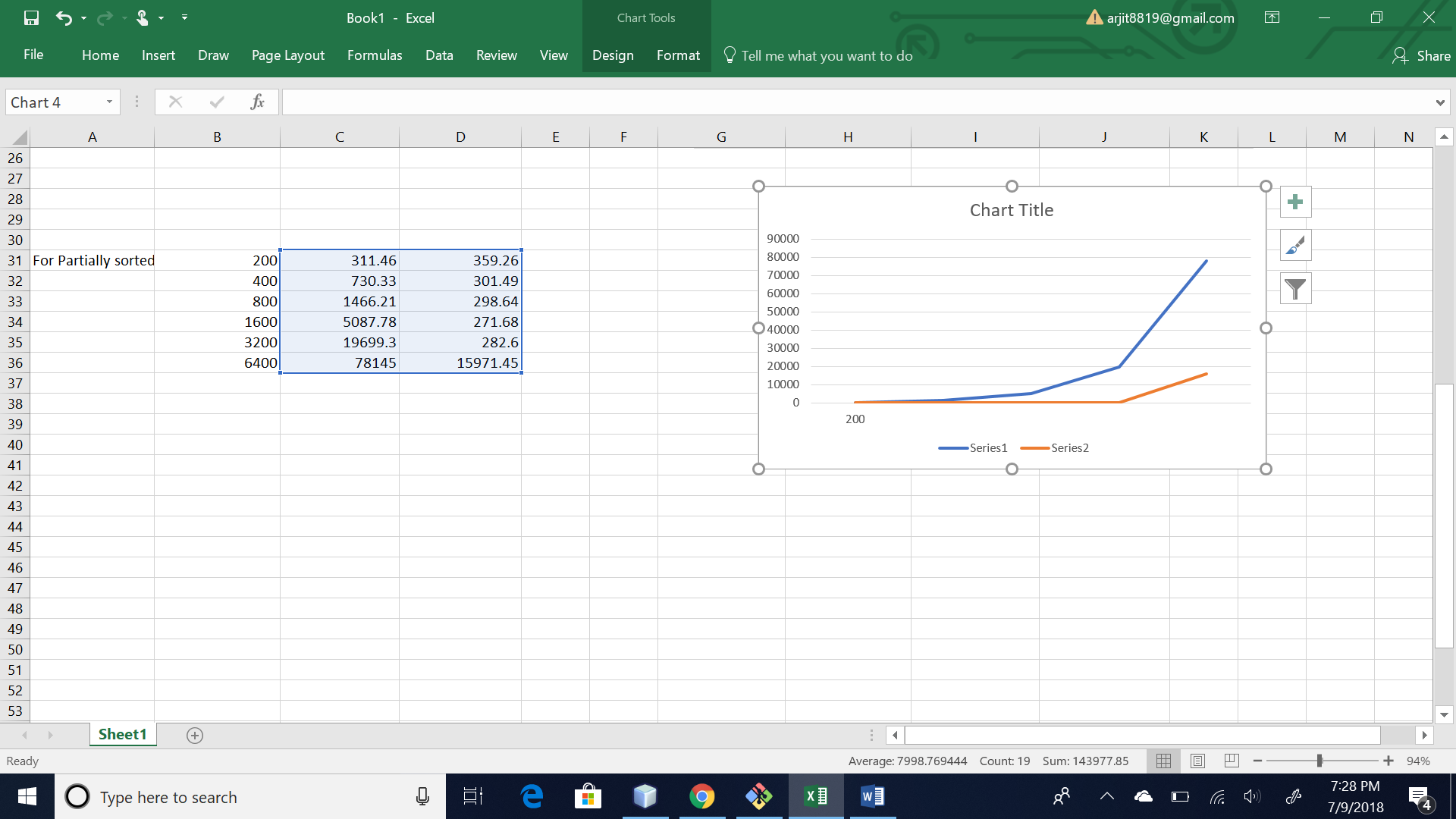
Insertion sort average time = 301.49.

Selection sort time is = 730.33.

For N = 800.

Insertion sort = 298.64.

Selection sort = 1466.21.



Below is the Reading for Random array:

For random array the time complexity for both the algorithm is same that is N^2

This can be observed from the readings

For N= 200

Insertion sort =313.83.

Selection sort = 476.14.

For N =400

Insertion sort =339.23.

Selection sort = 569.14.

