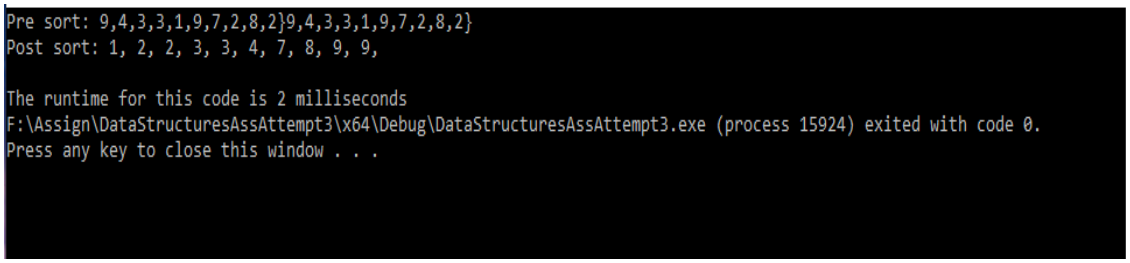
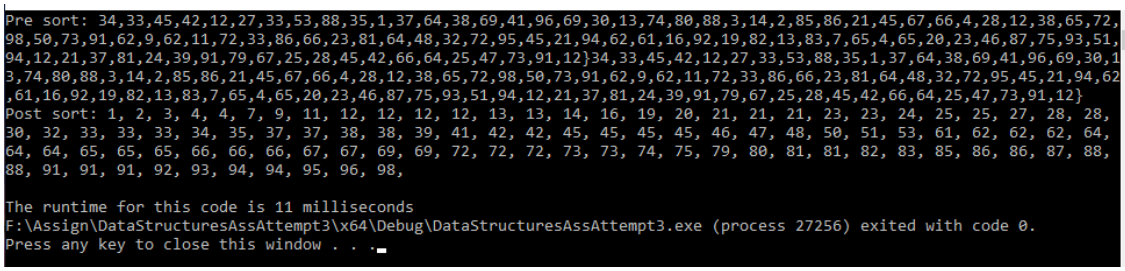
TASK OPTION 2: Comparing Sorting Algorithms
By: Nicholas & Tiffany

1) Please check our GitHub files in the following repository link to see our bubble sort, insertion sort, and selection sort code: https://github.com/HikariNoRyu/DataStructureGroupAssignment/tree/main
Yes, we decided to do those 3 basic sorting algorithms.

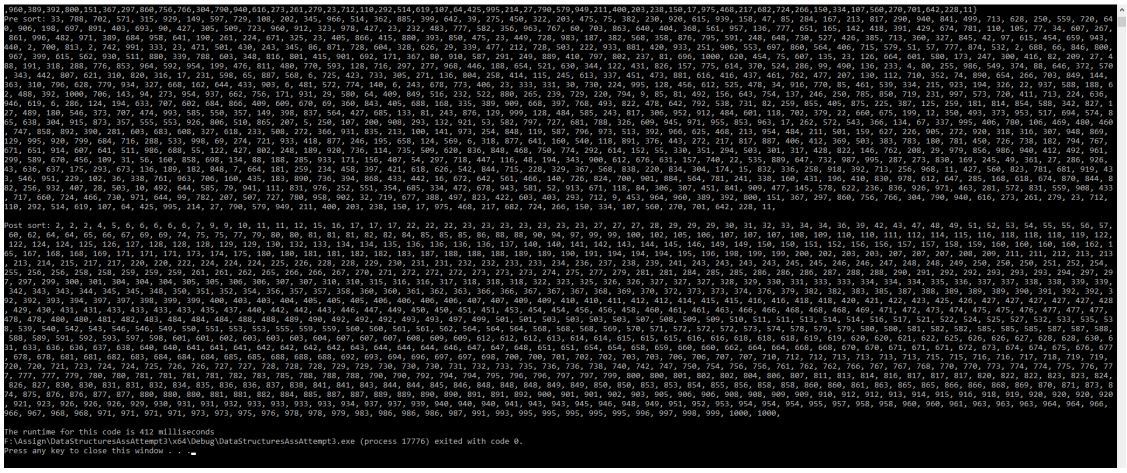
2 & 3) Insertion sort:



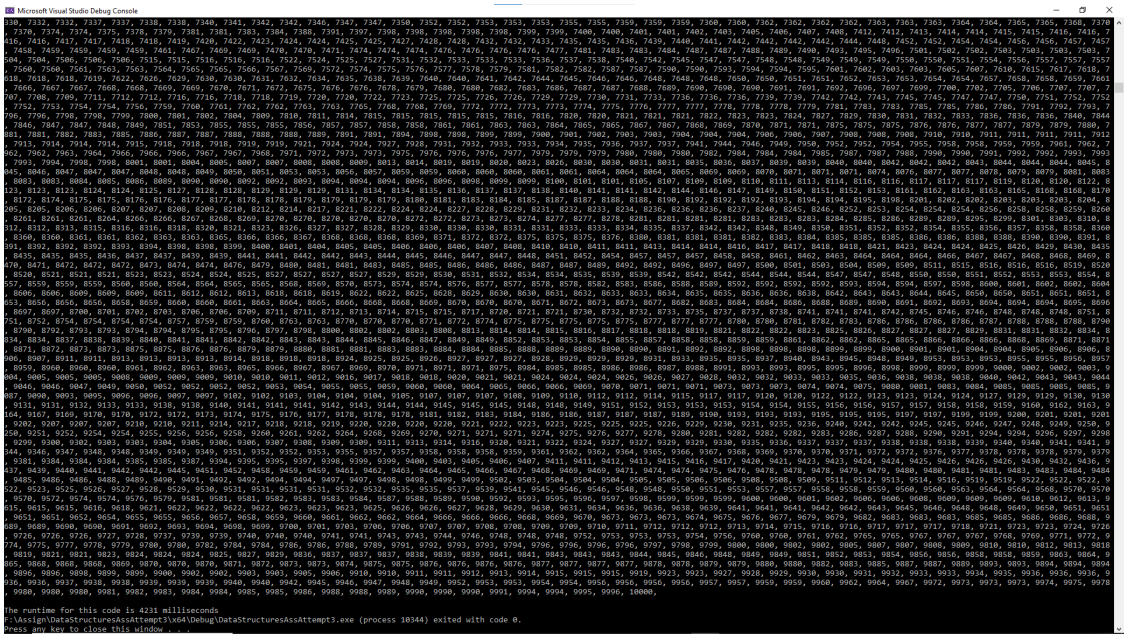
Running time: 2 milliseconds
input size: 10



Running time: 11 milliseconds
Input size: 100



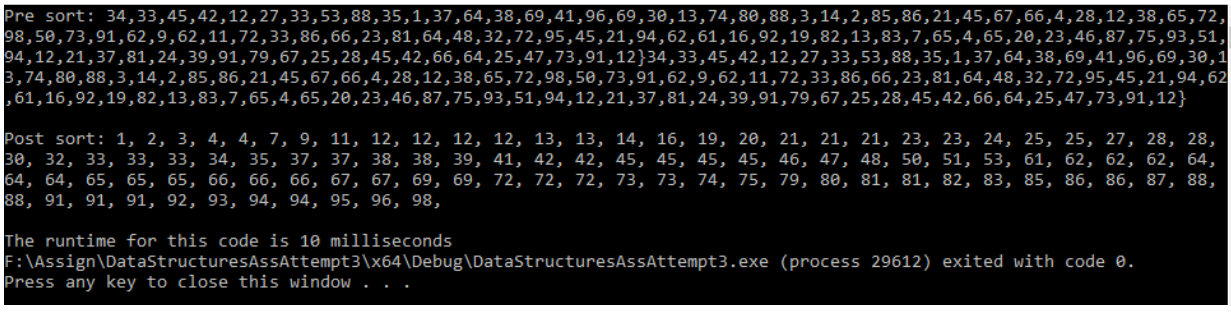
Running time: 412 milliseconds
Input size: 1000



Running time: 4231 milliseconds
Input size: 10000

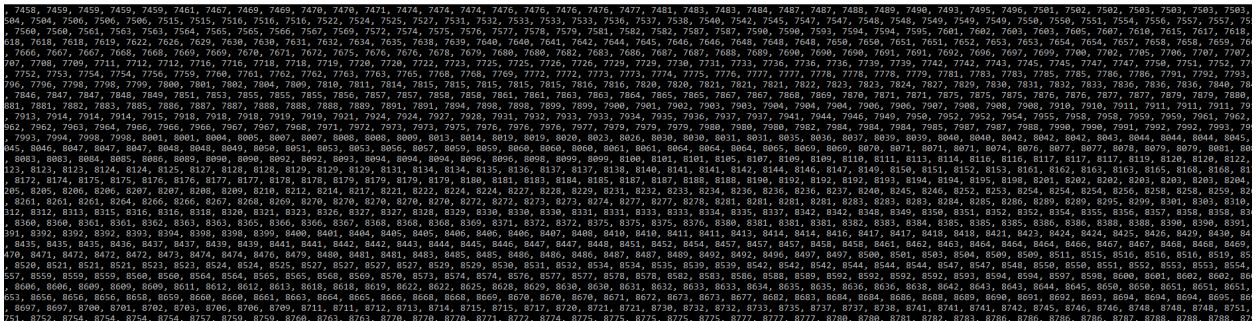
Selection sort:

Running time: 3 milliseconds
Input size: 10



Running time: 10 milliseconds
Input size: 100

Running time: 190 milliseconds
Input size: 1000



Running time: 2177 milliseconds
Input size: 10000

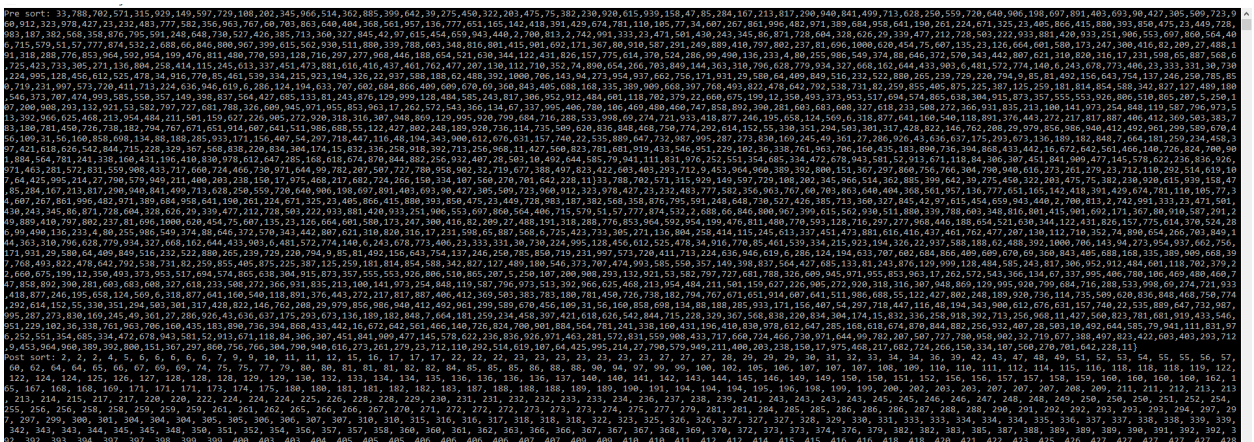
Bubble sort:

```
Pre sort: 9,4,3,3,1,9,7,2,8,2}9,4,3,3,1,9,7,2,8,2}
Post sort: 1, 2, 2, 3, 3, 4, 7, 8, 9, 9,

The runtime for this code is 7 milliseconds
F:\Assign\DataStructuresAssAttempt3\x64\Debug\DataStructuresAssAttempt3.exe (process 25096) exited with code 0.
Press any key to close this window . . .
```

Running time: 7 milliseconds
Input size: 10

```
Pre sort: 34,33,45,42,12,27,33,53,88,35,1,37,64,38,69,41,96,69,30,13,74,80,88,3,14,2,85,86,21,45,67,66,4,28,12,38,65,72,
98,50,73,91,62,9,62,11,72,33,86,66,23,81,64,48,32,72,95,45,21,94,62,61,16,92,19,82,13,83,7,65,4,65,20,23,46,87,75,93,51,
94,12,21,37,81,24,39,91,79,67,25,28,45,42,66,64,25,47,73,91,12}34,33,45,42,12,27,33,53,88,35,1,37,64,38,69,41,96,69,30,1
3,74,80,88,3,14,2,85,86,21,45,67,66,4,28,12,38,65,72,98,50,73,91,62,9,62,11,72,33,86,66,23,81,64,48,32,72,95,45,21,94,62
,61,16,92,19,82,13,83,7,65,4,65,20,23,46,87,75,93,51,94,12,21,37,81,24,39,91,79,67,25,28,45,42,66,64,25,47,73,91,12}
Post sort: 1, 2, 3, 4, 4, 7, 9, 11, 12, 12, 12, 12, 13, 13, 14, 16, 19, 20, 21, 21, 21, 23, 23, 24, 25, 25, 27, 28, 28,
30, 32, 33, 33, 33, 34, 35, 37, 37, 38, 38, 39, 41, 42, 42, 45, 45, 45, 46, 47, 48, 50, 51, 53, 61, 62, 62, 62, 64,
64, 64, 65, 65, 65, 66, 66, 66, 67, 67, 69, 69, 72, 72, 72, 73, 73, 74, 75, 79, 80, 81, 81, 82, 83, 85, 86, 86, 87, 88,
88, 91, 91, 91, 92, 93, 94, 94, 95, 96, 98,

The runtime for this code is 58 milliseconds
F:\Assign\DataStructuresAssAttempt3\x64\Debug\DataStructuresAssAttempt3.exe (process 12164) exited with code 0.
Press any key to close this window . . .
```

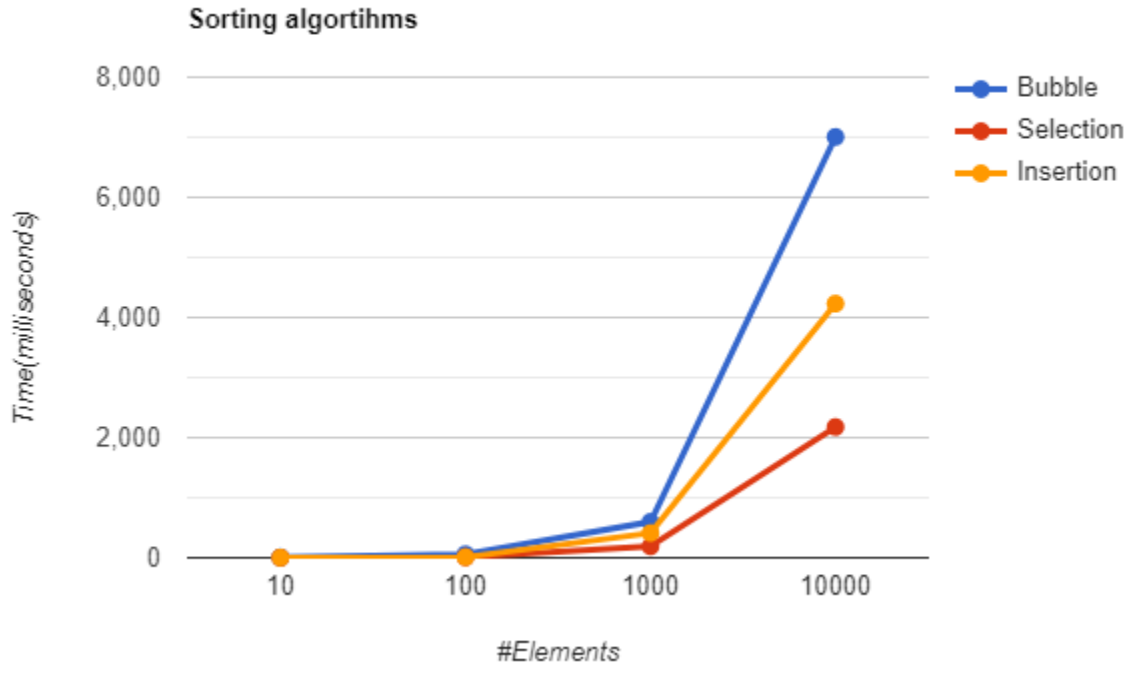
Running time: 58 milliseconds
Input size: 100



Running time: 598 milliseconds
Input size: 1000

Running time: 7006 milliseconds
Input size: 10000

4) Results of measurement of bubble, insertion, and selection sort running time comparison shown in the chart below:



5) Based on the chart, which of the three algorithms is the fastest/slowest?

The fastest one out of the three basic sorting algorithms is selection sort meanwhile, the slowest algorithm among the three is bubble sort.

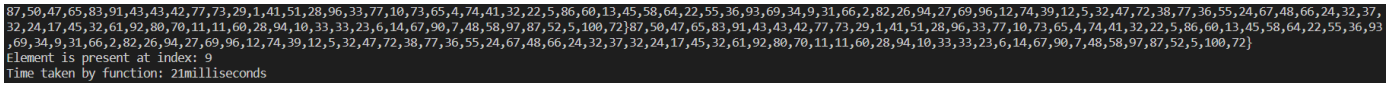
6) Identify each algorithm's complexity and discuss if the chart resembles the complexity function:

Bubble sort time complexity: $O(n^2)$ in the average and worst cases – and $O(n)$ in the best case.
Insertion sort time complexity: $O(n^2)$ in the average and worst cases - and $O(n)$ in the best case.
Selection sort time complexity: $O(n^2)$ in the average and worst cases - and $O(n^2)$ in the best case.

Extra:

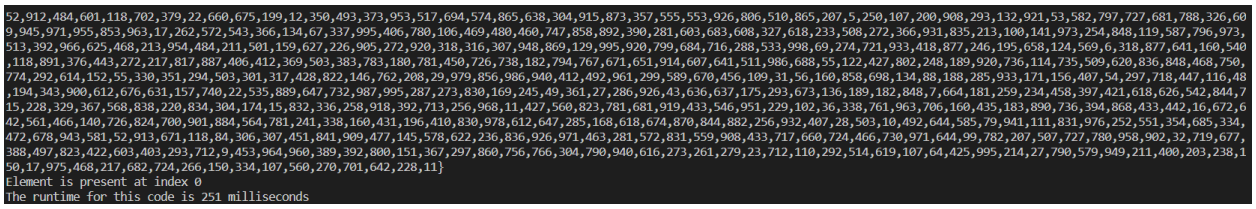
We also happened to think that our assignment was to do searching algorithms so here you go sir, here is our binary search and sequential search attempt:

Sequential search:



Running time: 21 milliseconds
The element being searched for: 77
Result: element 77 is found at index 9 of the array.

Binary Search:



Running time: 251milliseconds
The element being searched for: 33
Result: element 33 is found at index 0 of the array.