Quick Questions

1. **How many comparisons does insertion sort make on an input array that is *already sorted*?**

Constant	
Logarithmic	
Linear	X
Quadratic	

2. **What is a stable sorting algorithm?**

   Stable sorting algorithms maintain the relative order of records with equal values. That is, if there are two equal numbers to be sorted in the input, a stable sorting algorithm would not change the order of the two numbers - while that is not necessarily true in an unstable algorithm.

3. **What is an external sorting algorithm?**

   Algorithm that uses main memory during the sort.

4. **Identify 6 ways of characterizing sorting algorithms?**

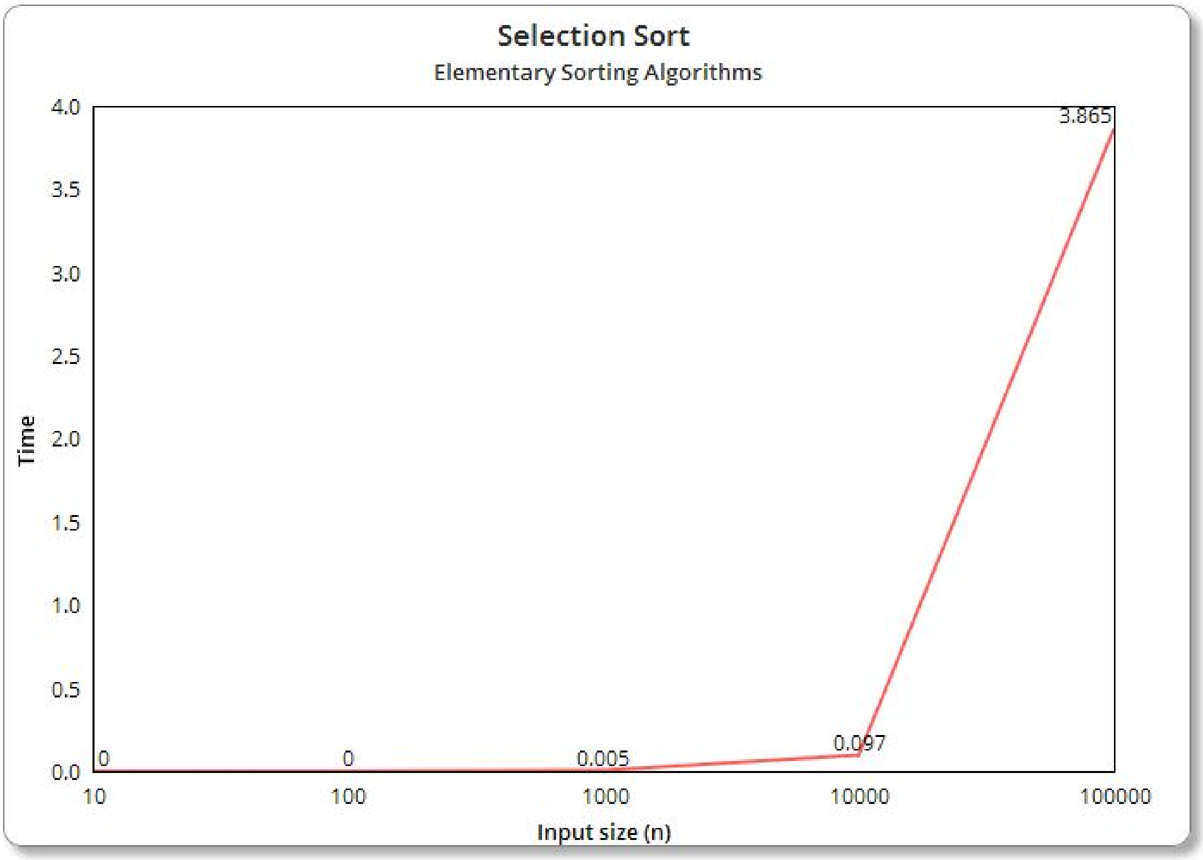
   1. Stable/Unstable
   2. External/Internal
   3. Time Complexity
   4. Space Complexity
   5. In place / Not in Place
   6. Recursive / Iterative

Algorithmic Analysis

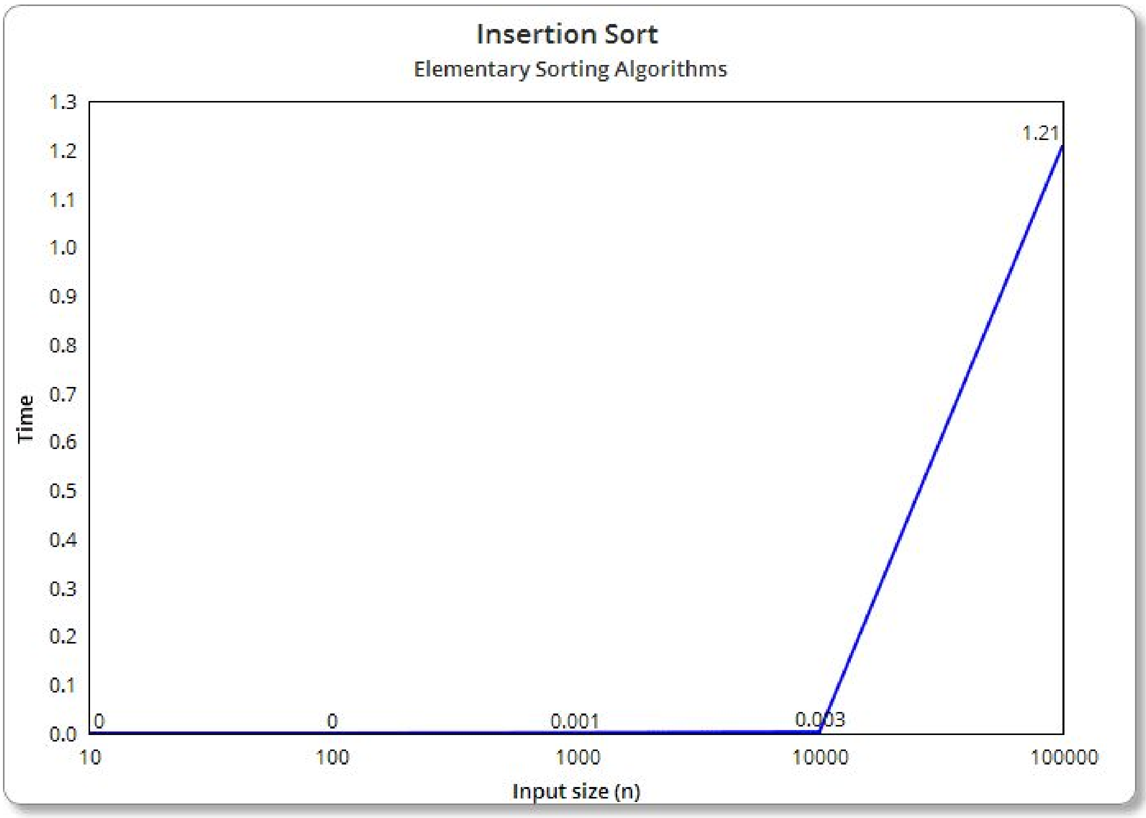
Empirical Analysis of the Elementary Sorting Algorithms

Input Size (n)	Time elapsed - Selection Sort	Time elapsed - Insertion Sort	Time elapsed - Stalin Sort
10	0.0	0.0	0.0
100	0.0	0.0	0.0
1000	0.005	0.001	0.001
10000	0.097	0.003	0.004
100000	3.865	1.21	0.007

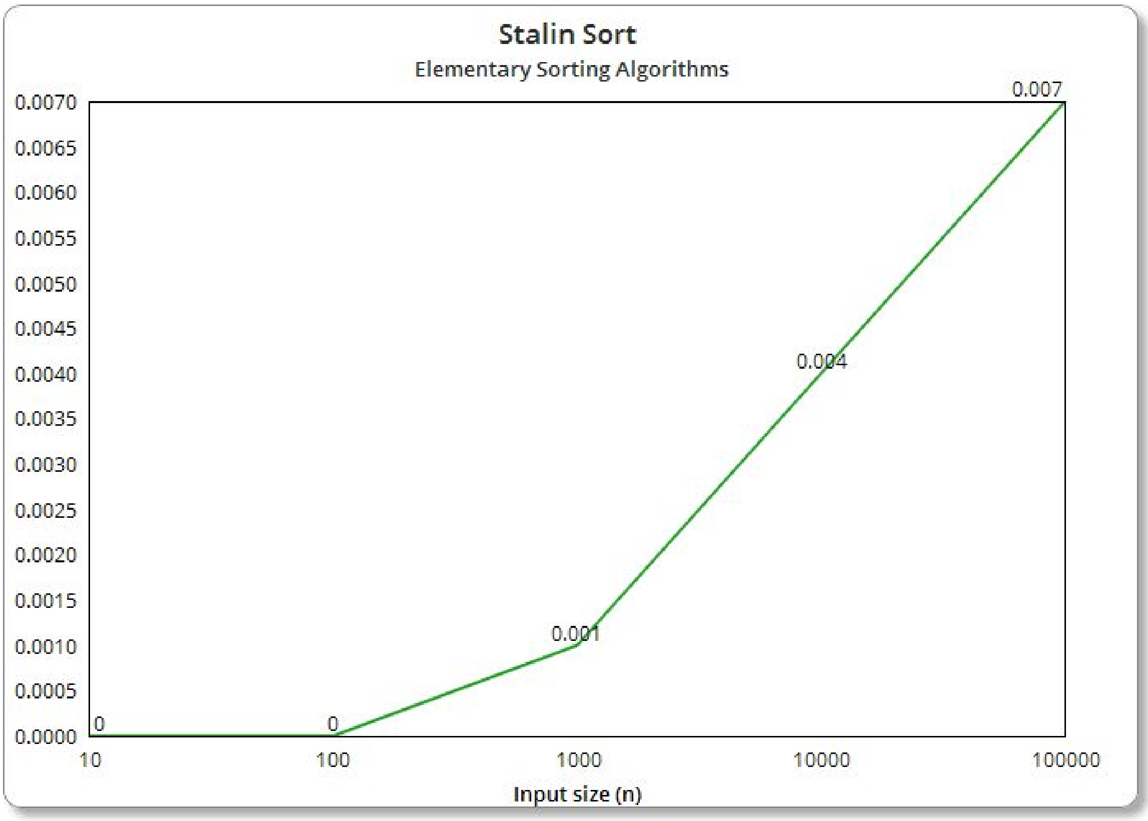
Selection Sort - Timing Graph



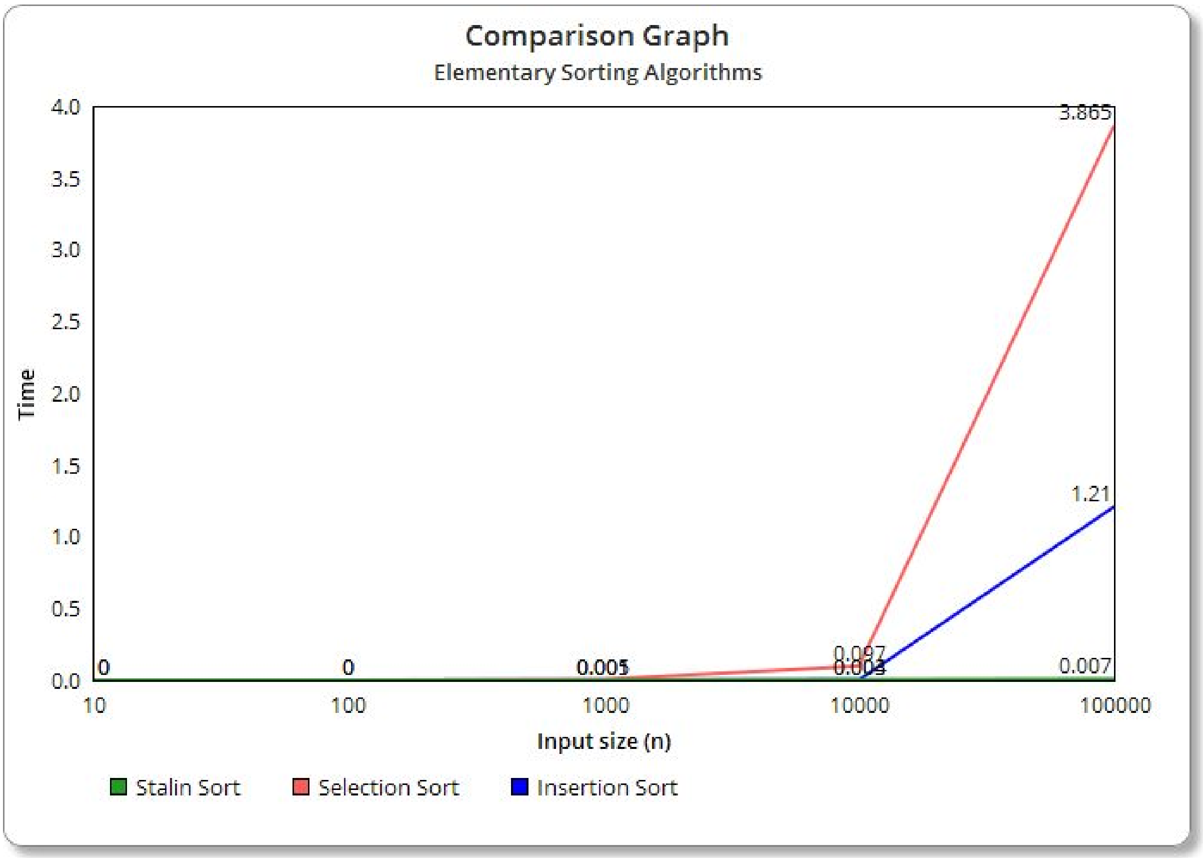
Insertion Sort - Timing Graph



Stalin Sort - Timing Graph



Comparison Graph



From the empirical results as well as the graphs above, we can see that neither selection sort or insertion sort are particularly efficient for larger input sizes.
I believe the time complexity for the two is quadratic - they both have two for loops - but, despite that, insertion sort seems to perform better on average.
The time complexity of the Stalin sorting algorithm is linear - a single for loop.