

Programming Assignment - I

Bubble Sort vs Insertion Sort

Plots:

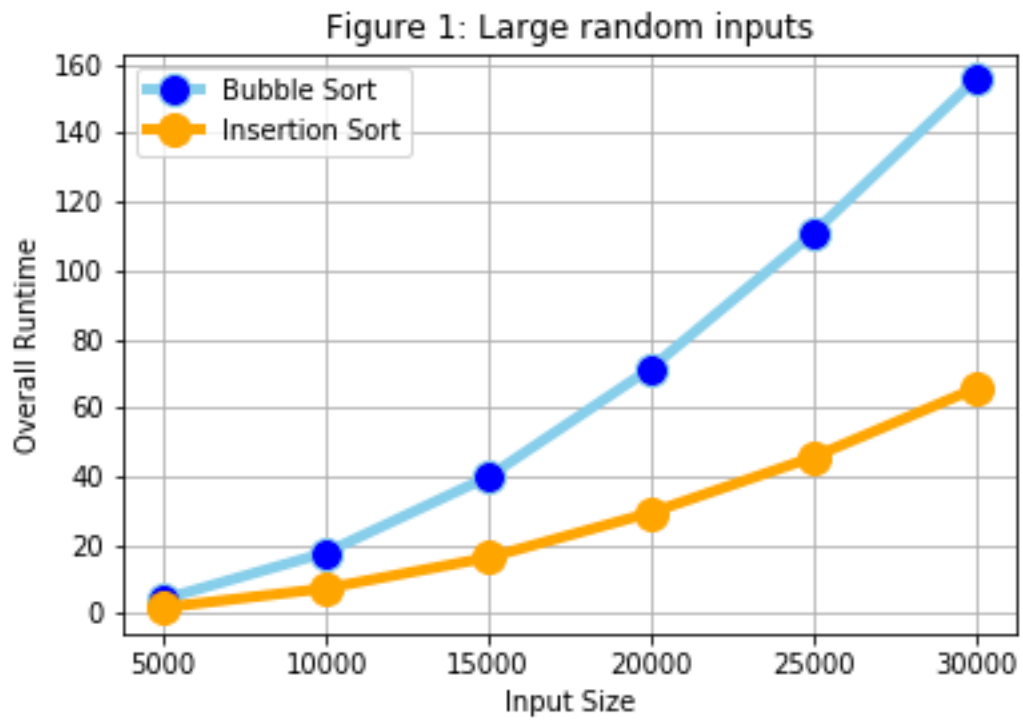


Figure 2: Non-decreasing inputs

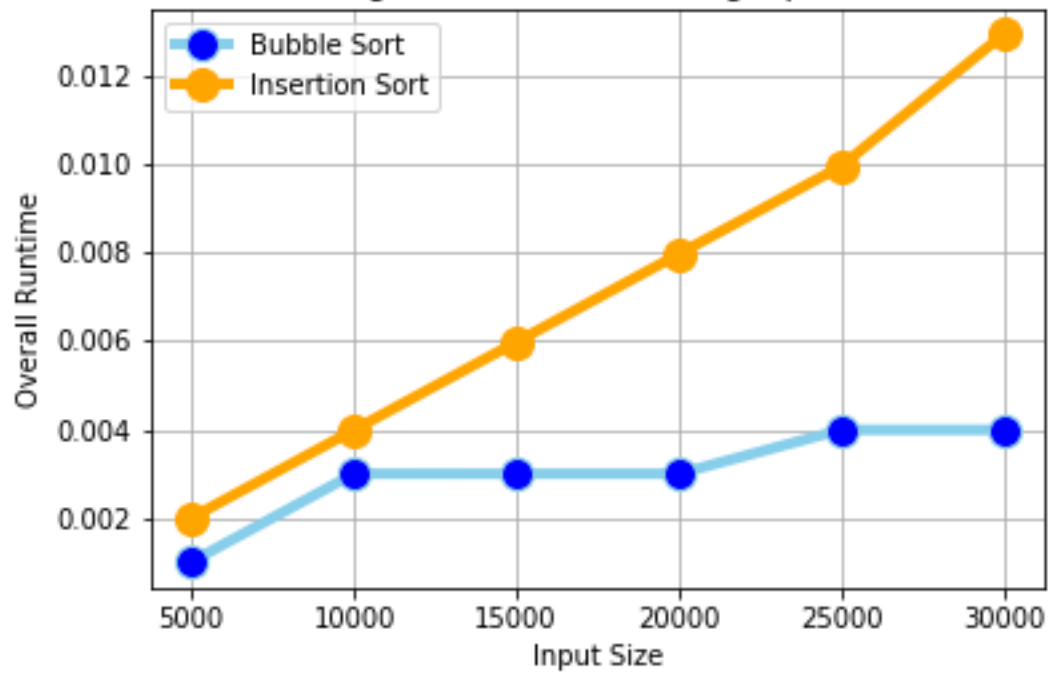
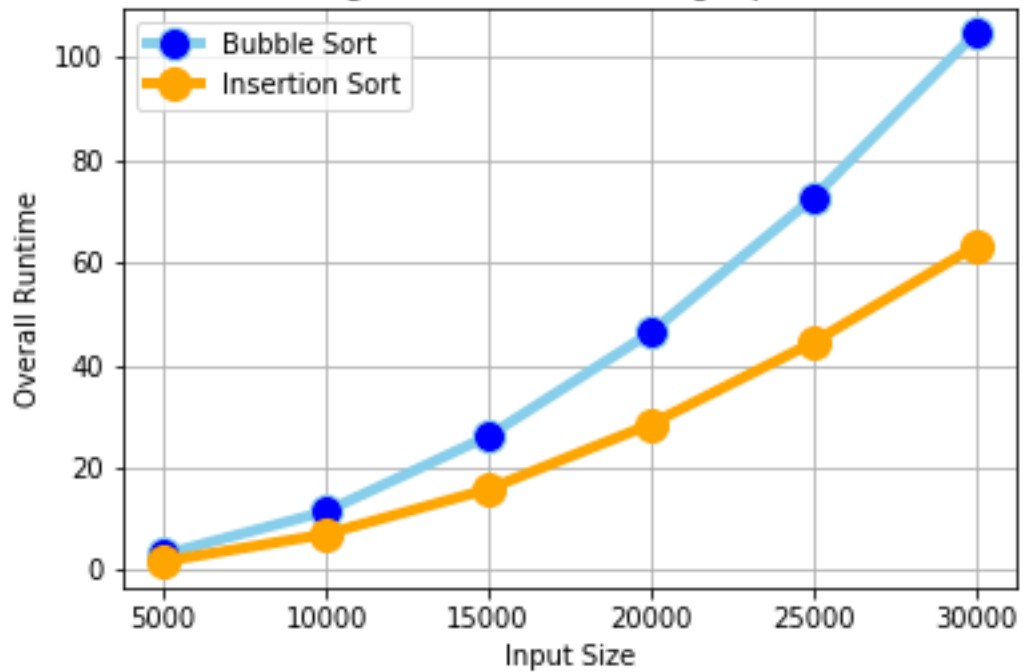
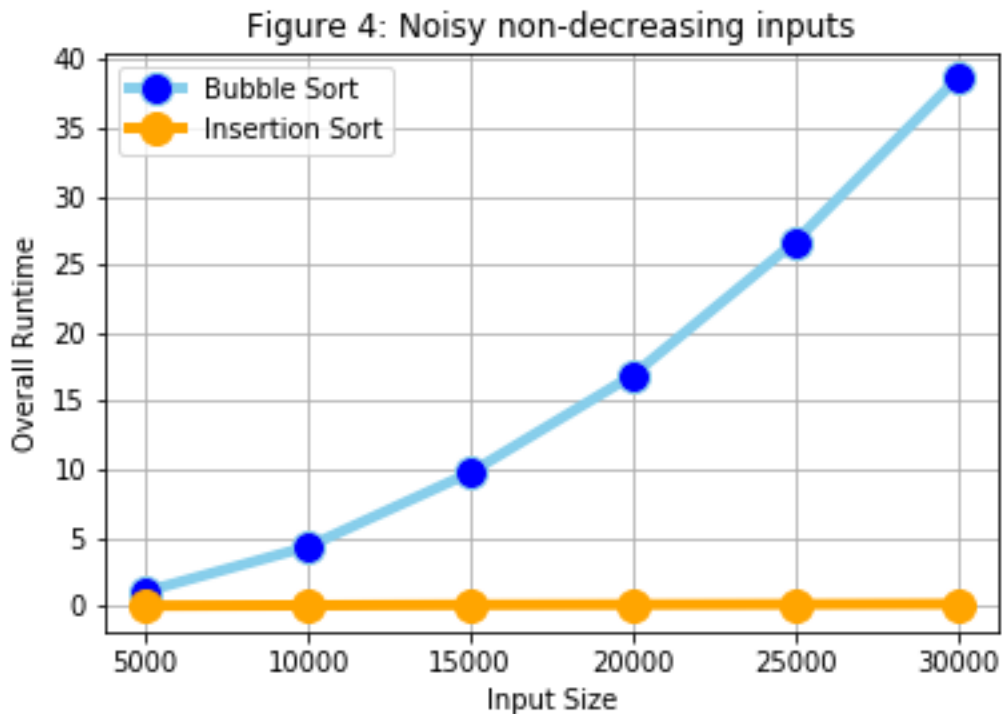


Figure 3: Non-increasing inputs





Input 5 overall runtimes:

Bubble Sort: 45.89 seconds

Insertion Sort: 20.09 seconds

Explain your choices:

For this assignment, I have used python for programming. I have generated random inputs using python NumPy library. Also, saved and plotted graphs using matplotlib library. To record time, I used python time library. For storing the data for plots, I created two dictionaries, one for bubble and, one for insertion sort, whose key is input type and values are list of runtimes for each size.

Conclusions:

1) Does the first plot reflect this? How do the two algorithms compare in terms of the running time?

Both the algorithms have time complexity of $O(n^2)$. As the input size increases, runtime of algorithms also increases. Insertion sort performed better than bubble sort for randomized inputs. Because, bubble sort checks and swaps each adjacent element if necessary, whereas insertion sort just put element in its correct position in a sorted portion of array.

2) How about the second plot? Do you think this one is quadratic? Why do you think it looks the way it does?

Since the input is already sorted, the algorithms don't have much computations to do. Best case complexity of bubble sort and insertion sort is $O(n)$. Hence, our plot is linear and not quadratic.

Bubble sort exits the loop right after one iteration because of checks implementation. Insertion sort has some value assignments at each iteration hence it takes slightly more time than bubble. Both the algorithms run within milliseconds.

3) How does the third plot compare to the first and second?

First and third plots are almost similar and insertion sort performed better in them. Slope and curves look almost the same for these plots.

Second plot is linear whereas third plot is quadratic. Bubble sort performed better in second whereas insertion sort performed better in third plot.

4) What about the fourth plot? Can you explain the behavior?

Fourth input is almost sorted. Insertion sort works better than bubble since insertion sort does very less swaps as compared to bubble sort. In insertion sort elements are moved into the sorted section, while in bubble sort the maximums are bubbled out of the unsorted section.

5) What about the fifth input? Note that the total input size is greater than all other inputs combined; why the algorithms can handle it efficiently?

The algorithms handle it efficiently because, for the smaller array sizes, time and space complexity is very less as compared to large array sizes. Bubble sort took approx. 45 seconds to run whereas insertion sort took 20 seconds.

6) Summarize in which situations these algorithms can be used in practice. Which one would you prefer?

In general, bubble sort is slower than insertion sort. Bubble sort is preferred when you want to sort n elements since it bubbles out larger elements at each iteration. I would prefer insertion sort since it performs well in most of the cases.