CS 3310
Dr. Gupta
Mariia Kravtsova
SLC Report
November 13, 2016

Assignment 4 implementation has the following classes:
BubbleSortAL - Bubble Sort on built in Array List implementation
InsertionSortAL - Insertion Sort on built in Array List implementation
SelectionSortAL - Selection Sort on built in Array List implementation
MergeSortAL - Merge Sort on built in Array List implementation
LinkedList - Implementation of linked list and Bubble, Insertion, Selection and Merge sorts.

This code is run on a 2.5GHx i7, with 16 GB RAM and GeFroce GT 750M 2GB MacBook Pro.

At first, let us compare the complexity for the basic methods of Linked Lists vs Array Lists. It goes as follows:

	Linked List	**Array List**
get item	O(n)	O(1)
add item	O(1)	O(1)
remove item	O(n)	O(n)

Since this sorts are the basic compare then swap sorts, the theoretical vs empirical analysis is fairly close to each other. Granted, I did not successfully completed merge sort for the Linked List so my analysis is mostly an approximation.

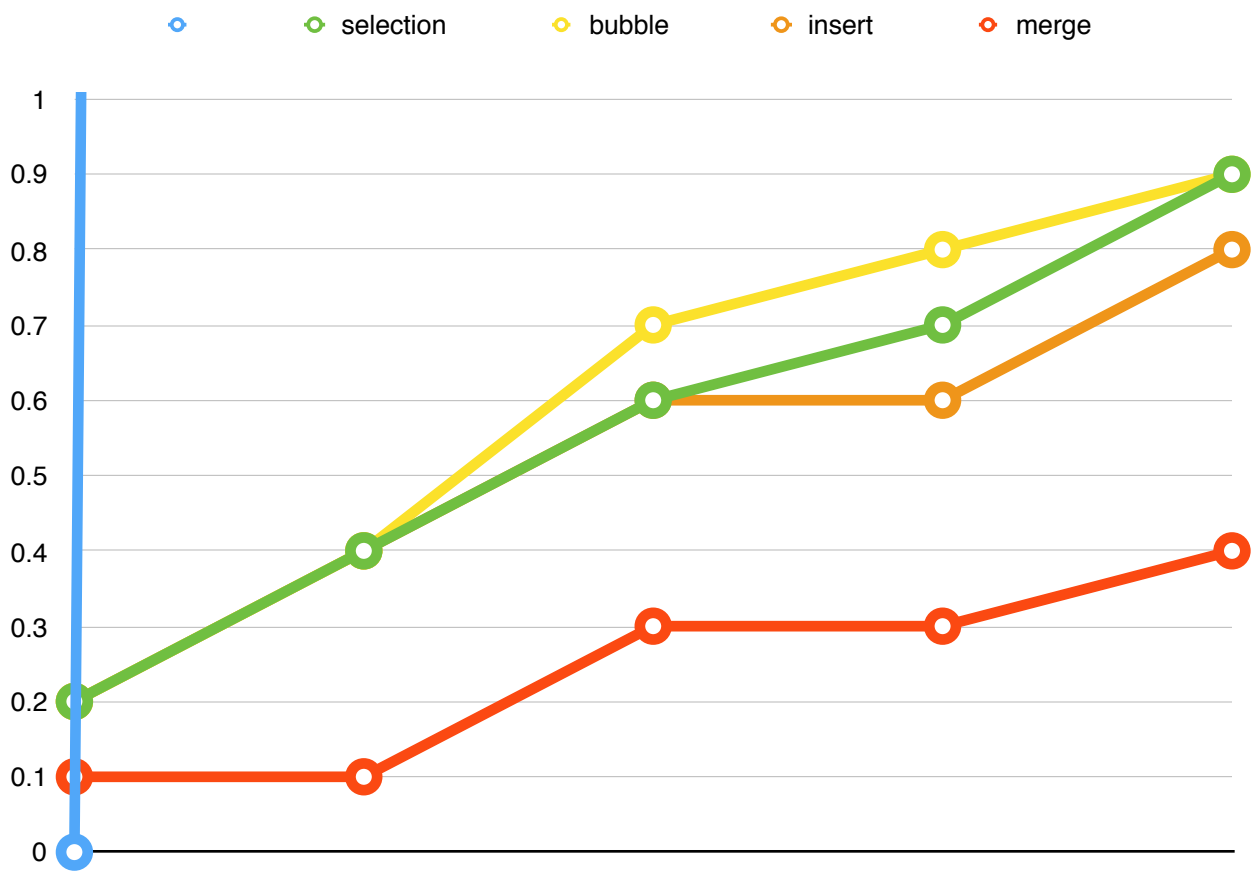
The BubbleSort on the Array List and on the Linked List is at a n^2 growth function. Which comes at a no surprise since swapping methods are very easy at this point. So the only n are in the two for loops. Yet again it proves how slow this sort is.

The SelectionSort on the other hand is a bit more complicated, but it is still running close to the n^2 function, which is what expected. Although when the data becomes smaller the function slows down.

The Merge Sort appears to be growing slower than the rest, but it is still not at n log n which we would expect from the theoretical one, but the swap is a bit different so it is to be expected slower on the bigger scale.

The Insertion Sort is n^2, it performs better compared to the rest of the sorts when the data is bigger, which is not what I expected. Since insertion sort should be better on the smaller data sizes.

The Y axis is represented by Average in milliseconds and the X axis is represented by the list size. I had a separate file than the one we were provided, in which I increased the data, but tested it on integers instead of strings, because this was easier to generate.



In conclusion, I believe that yet again we have proven that the sorts are not as efficient as we would think of them to be in theory. And Although the functions appear to be more linear we can see by the numbers that if they keep their tendency growth it will be growing into a parabola. Unfortunately, my computer could not run more data within the reasonable time for me to submit the assignment.