Name: Sena Yevenyo
Date: September 11, 2019
Lab Section: Section 1.

1. Introduction

The purpose of this lab is to collect and compare the run times of insertion sort, linear sort for both sorted and unsorted lists, binary search and linear search run times.

The lab manual instructed the creation of a StringContainer class which contains Strings, a RandomStringGenerator which generated String as digits based on a seed and lengthOfString input and an ExperimentController to be the main program that runs and generates the data needed for the conclusion of the lab.

No assumption was made in this lab.

2. Approach

No data structure was implemented for this lab. Data class objects used were Java ArrayList as instructed by lab instruction manual. Java Arrays class objects were also used in this program.

StringContainer class main function is to store Strings.

RandomStringGenerator class main function was to generate a series of Strings of a specific length and in only digits.

ExperimentController class is the main program that runs to produce the average run time for each method of the ExperimentControl class.

3. Methods

Runs are not hard coded into program and thus, a choice of the user. Parameters used by the program, as instructed in the lab manual, is the file name for data written by program.
NumberOfItems data to control the number of data inputs to a StringContainer objects and number of items to sort.

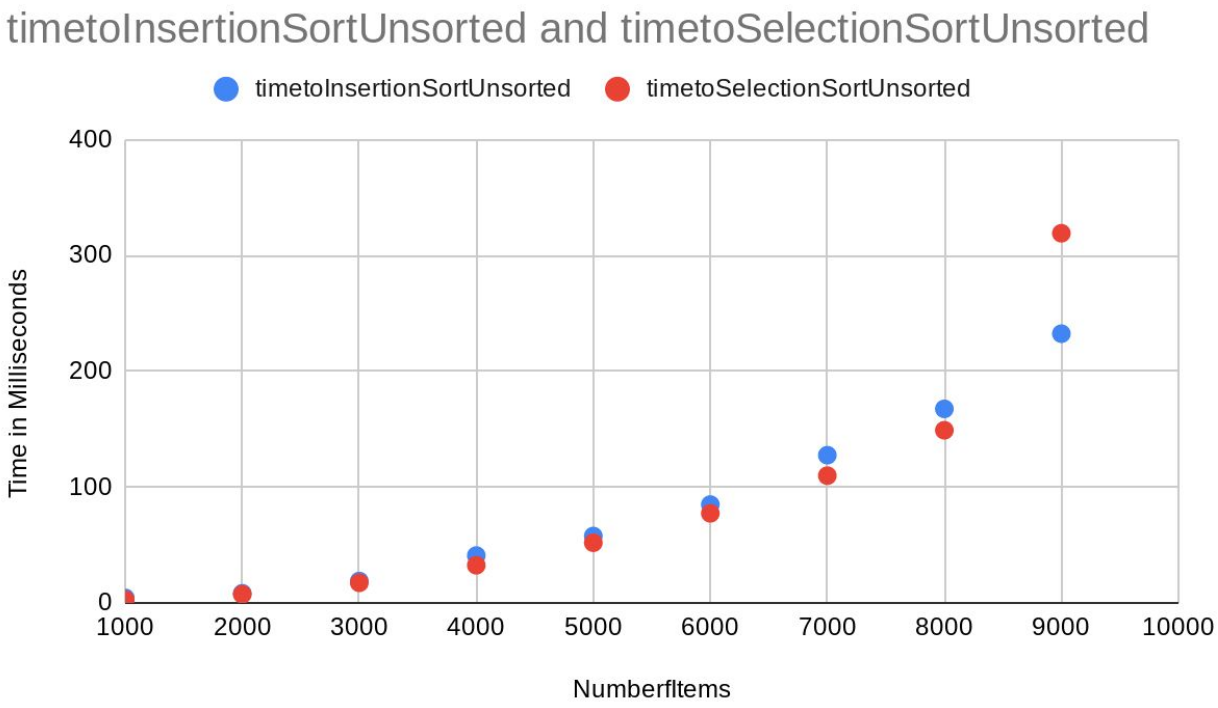
Name: Sena Yevenyo
Date: September 11, 2019
Lab Section: Section 1.

4. Data and Analysis

Number of runs for this data is 15.
Number of items sorted and added to StringContainer objects are 1000 to 9000 inclusive.
Seed used was an arbitrary number of int 1000 and strLength of 5 for RandomStringInteger class.

Data generated is attached as a .csv file with name “output.csv”.

Sorts.
Insertion Sort and Selection Sort Unsorted

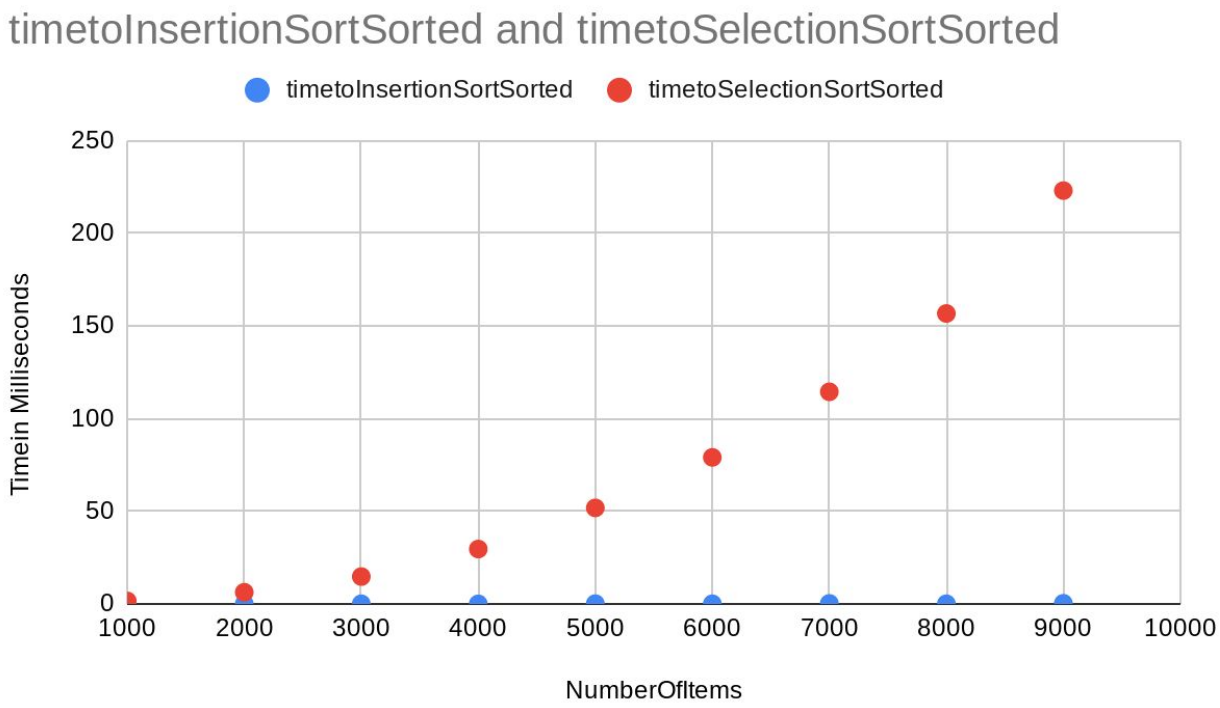


Insertion and selection sort in theory should have similar runtime for the same N number of items sorted for unsorted list. (O(N^2) for selection sort and O(N^2) for insertion)

Name: Sena Yevenyo
Date: September 11, 2019
Lab Section: Section 1.

This graph follows a similar trend with values very close to each other. Values diverge from each other as N increases.

Insertion Sort and Selection Sort for a Sorted List



Insertion and selection sort in theory should have different runtime for the same N number of items sorted. This graph follows a similar trend with values diverging from each other as N increases.(O(N^2) for selection sort and O(N) for insertion)

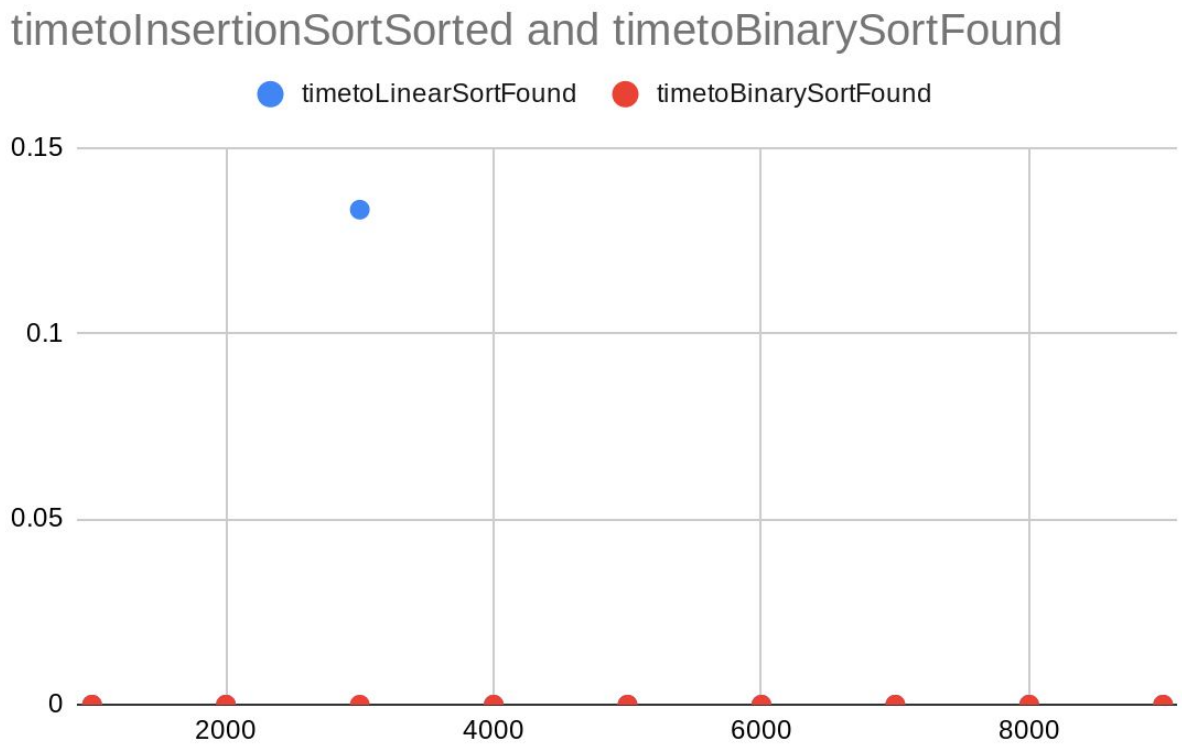
There is a bit of an anomaly here. Though insertion sort is supposed to increase less faster than selection, it is supposed to increase as N increases.

Name: Sena Yevenyo
Date: September 11, 2019
Lab Section: Section 1.

The straight line gotten in this graph may have been a result of the smaller N number of inputs.

Searches

LinearSearch and BinarySearch for a String key in the list

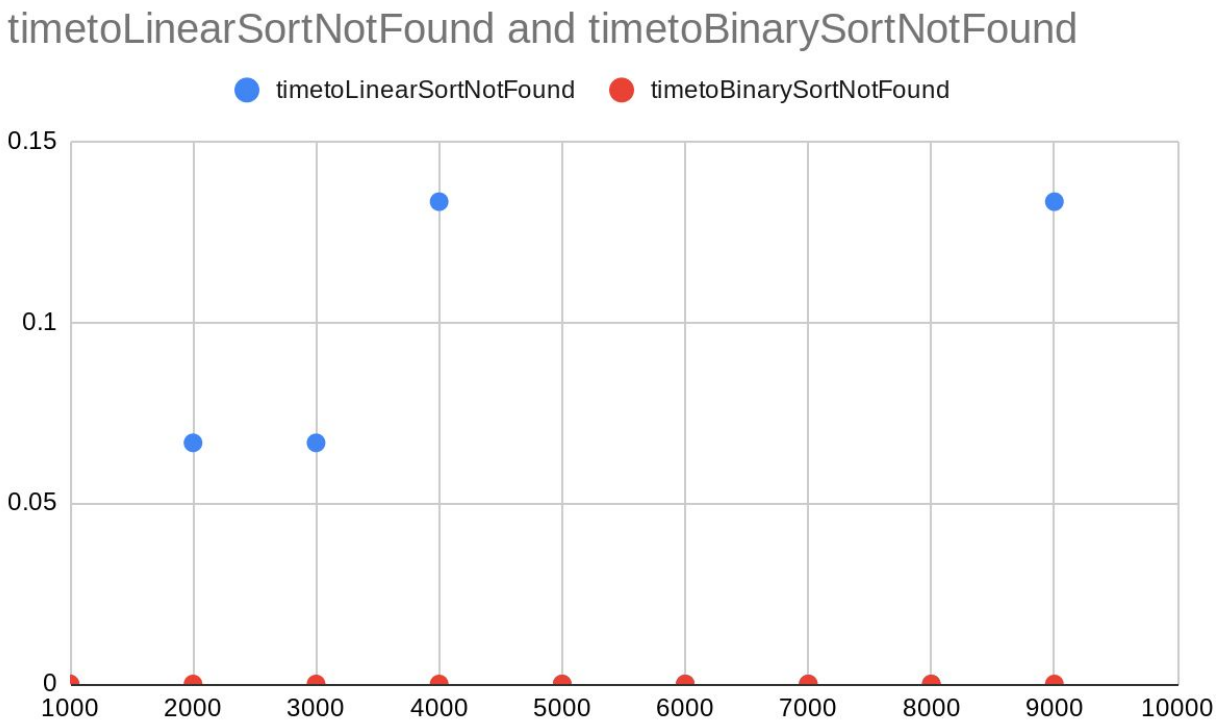


Linear search and binary search have different runtimes in theory (O(logN) for binary and O(N) for linear search) but the values in the graph show a similar trend for both searches.

Reason for this might be how the algorithm was implemented in the code. The worse case scenario runtime was probably not achieved due to a randomness in choosing the String key to be searched. Thus one run might choose a String at the beginning of the StringContainer object but in another run, might choose one at the end.

Name: Sena Yevenyo
Date: September 11, 2019
Lab Section: Section 1.

LinearSearch and BinarySearch for a String key not in the list



Linear search and binary search have different runtimes in theory (O(logN) for binary and O(N) for linear search).

Here, since key was not present in StringContainer, there is more divergence between linear search and binary search but not a huge difference.

Reasons for this is most likely the value of N's that were used in these experiments. Low value of N's tend not to completely show the correlation between run time values.

5. Conclusion

Name: Sena Yevenyo
Date: September 11, 2019
Lab Section: Section 1.

All classes created performed as they should. Data obtained could not completely show the big-O runtime notation for algorithms due to low N values and implementation of algorithms.

No major difficulty has met in this lab.

6. Reference

No references were used in this lab.