Matthew Curry – February 10, 2019 – CS 150 Lab 2
Lab Notes

Introduction

This lab featured two classes in addition to a unit testing class. The first class dealt with the object “RandomIntegerContainer” which is an arrayList of integers with various methods defined for adding elements to the empty arraylist and ordering the elements of the empty arraylist. The second class was an “experiment controller” class which ran the various methods of the RandomIntegerContainer for a set number of items and a randomly generated set of integers. Finally, a unit testing class existed to test the various methods used on the RandomIntegerContainer.

Method

The first class defined the object RandomInteger Container. This object is an empty arraylist of integers that includes various methods to add integers to the container or to reorder items in the container. The first of these methods is the “addToFront” method which takes an Integer input and adds it to the front of the integer container. Next, there is an “addSorted” method which takes an item and adds it to the integer container in the proper numeric order. Next, there is a “bubbleSort method which bubble sorts the entries in the random integer container. Then, there is a bubbleSortUnique method which performs the bubble sort on the elements in the integer container before removing repeated values. Lastly, there is a swap method which switches elements in the container and is used by the bubble sort method and the addSorted method.

The other class is called “ExperimentController”. This class features the main method which creates an instance of this class and then calls a series of methods that call the methods for the RandomInteger container and records the time it takes to run these methods.

Unit tests

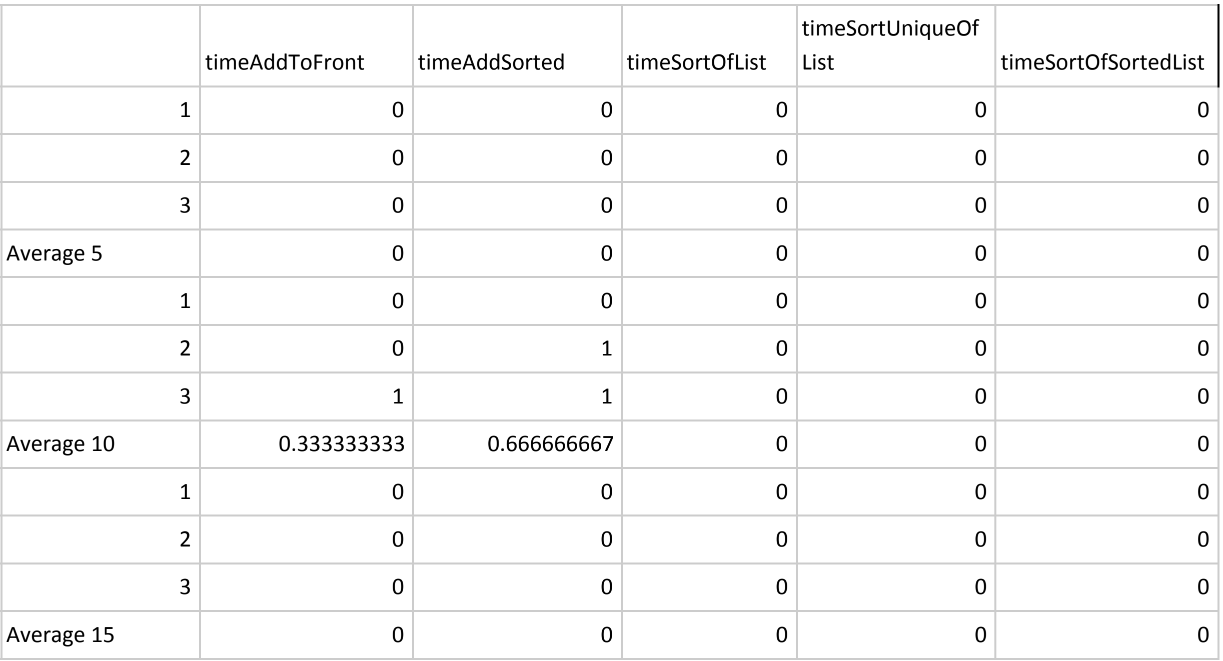
Unit tests were run on the methods in the RandomInteger container. Initially, there was an issue with the null point exception because there was a “void” in defining the constructor method. After this was fixed, the tests of the different methods ultimately all passed.

Experiments conducted

The experiment controller class ran every method for the random integer container and recorded the time it took for each to run. Data was gathered on adding 5, 10, and 15 items to the integer container through the various methods. 3 trials were conducted for each amount of added items, each with a different seed being used for the random generator to generate the set of random integers to be added. The seeds used were 20, 50, and 1000. The average for a given number of added items across these three seeds was calculated as well.

Results (analysis of the data)

The results for the times of these different methods is shown below:



	timeAddToFront	timeAddSorted	timeSortOfList	timeSortUniqueOf List	timeSortOfSortedList
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
Average 5	0	0	0	0	0
1	0	0	0	0	0
2	0	1	0	0	0
3	1	1	0	0	0
Average 10	0.333333333	0.666666667	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
Average 15	0	0	0	0	0

As all the time amounts for the methods timeSortofList, timeSortUniqueOfList, and timeSortOfSortedList were all 0, I did not include a graph.

Conclusion

Based on this data, we can conclude that most of these methods occur virtually instantaneously as we almost always had a value of 0 for the time it took for the method to run. The timeAddToFront method and the timeAddSorted method were the only methods to register non zero values, so it may be appropriate to conclude that these methods take slightly longer

Trouble report

The main issue dealt with the null point exception error previously addressed in the unit testing section

References

https://www.youtube.com/watch?v=6Gv8vg0kcHc

https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html