Abiola Gabriel Olofin

CS 150-02

Professor Falconi

02/09/2020

Lab 2 Report

Introduction

In this lab, we were asked to create a random generator that generates a random number from a string and inputs those numbers into an arraylist using different methods that would insert those numbers in different orders In other words, we are generating number and based on the order we want the numbers to be in, we use different methods to put those numbers in that order.

Approach

When I was designing the program, I only created one class that ran the entire program called ExperimentController and created the RandomIntegerContainer class to make the arraylist and to make the methods to add the numbers to the arraylist. I imported the java.util.* in both classes which imports all the libraries in Java to make it easier to call certain classes and methods of different classes. In the ExperimentController class, I created 6 methods to test the time that it takes each method in the RandomIntegerContainer class to complete the task. I also created a main method that would run each of those methods to generate the answers needed. In the main method, I tested each of the methods through the use of while loops in order to test how different lengths of arraylist are populated and how much time it takes depending on

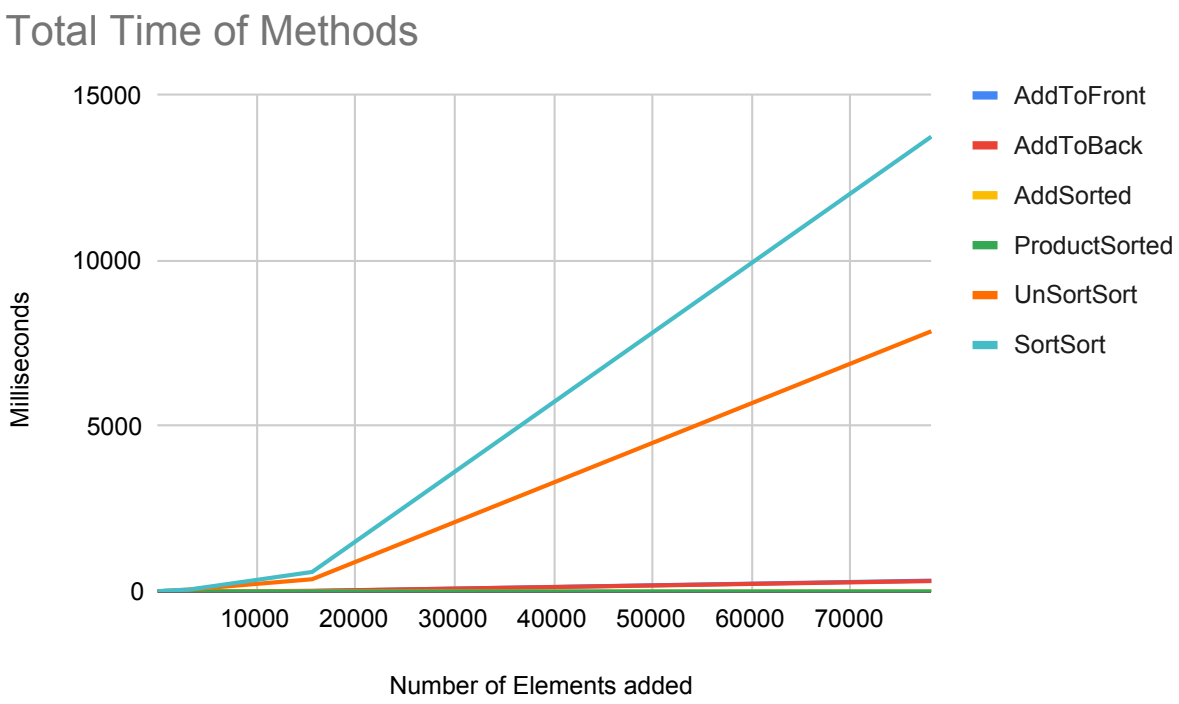
the method. In the RandomIntegerContainer class, I created 6 methods that added the values to the arraylist in a specified way and I created a method that turned the arraylist into an array with the same values as the arraylist.

Methods

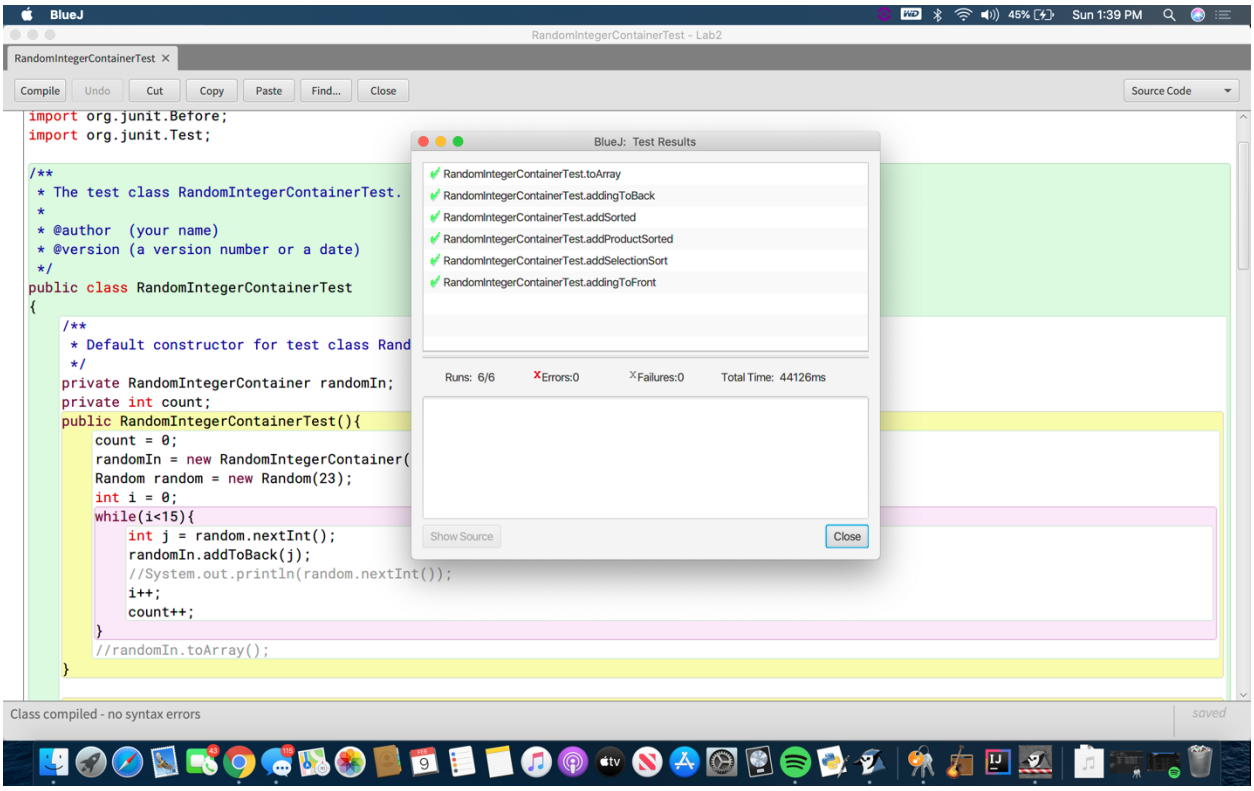
The methods I wrote in the RandomIntegerContainer class are addToFront, which adds a number as the first element in the arraylist moving all the other elements up by one position, addToBack, which adds a number to the back of the arraylist, addSorted, which adds a number and sorts that number into the arraylist, productSorted, which multiplies the first element in the arraylist by a number and sorts that new number into the arraylist and removes the previous first element, selectionSort, which sorts the arraylist using the selection sort algorithm, and finally toArray, which turns the arraylist into an array with the same values as the arraylist. I unit tested this class to assert that those methods in the class actually produced the right thing based on what the right output should be. Next, the methods I wrote in the ExperimentController class methods that returned the total time it takes the RandomIntegerContainer object to use each method in the RandomIntegerContainer class.

Data

I collected data to graph by running multiple trials of the timed methods to see how long it takes the computer on average to complete the methods and which methods took more time to complete and how much time it takes certain amounts of arraylist size to take to complete those methods.



Number of Elements added						
	AddToFront	AddToBack	AddSorted	ProductSorted	UnSortSort	SortSort
5	0.1	0	0	0	0	0
25	0	0.1	0	0	0	0.2
125	0.1	0.2	0	0	1.4	2.3
625	0.1	0.8	0.1	0.3	16.1	27.4
3125	1.2	1.9	0.8	0.3	60.4	58.3
15625	17.9	22.1	1.5	2	373.7	593.9
78125	330.8	318.7	5	4.6	7870.2	13753.9



Conclusion

Overall, this program was successful, and it taught me how to Junit Test correctly and to use the assert statement to prove that my code works.

References

ArrayList API

Collections API

Random API