

Erick Daniel Garcia
U0823062

Analysis Document for Assignment 03

My partner for this assignment was Joshua Shipley U# 0919708. I am the one who submitted the source code. While working with Joshua I noticed that he has a solid grasp on programming in Java, but sometimes struggles with concepts. He is somewhat of the opposite of me, As I have struggle programming in Java ,but have a good grasp of the concepts. We met around four times. In two of those I was the leader and would go over concepts,write code, while he would supplement my lack of Java knowledge when it was needed. The other two times he was leading and would write code while I would help with concepts. In all of our meeting we debugged together. I wouldn't mind working with him again, my only problem would be finding good times to meet because we both had have very busy schedules.

If we used an ArrayList instead of a normal Array the assignment would've been A LOT easier. ArrayList being a part of the collections library has built in methods that can be used when adding removing and sorting elements in the ArrayList. It also has a built in clear function and Iterators for objects inside an ArrayList.

I believe that the contains method which runs binarySearch has a complexity of Log base 2 of N or rather $O(\log(N))$.. The test we made which runs 10 experiments of searching for a random number in different sizes of Arrays consisting of 2^{11} , 2^{12} , 2^{13} And so on until 2^{20} elements in the array. Each experiment ran a million times and I took the average of each experiment and plotted that. Which was a logarithmic graph.

For the add function I predict a $O(N)$ complexity. The best case scenario is $N + \log(n)$ but the worst case will just have $O(N)$ because in a worst case scenario we will search through every element in the array making it a linear function. The tests I ran for the add function was similar to the last one. I performed 10 experiments consisting of Arrays containing 100,000 elements and added an element ,timed it and ran that 100,000 times. After taking the average of the 10 experiments I graphed the results and got a linear graph the looks like $O(N)$.

This assignment took us around 12 hours. Thanks for your time.