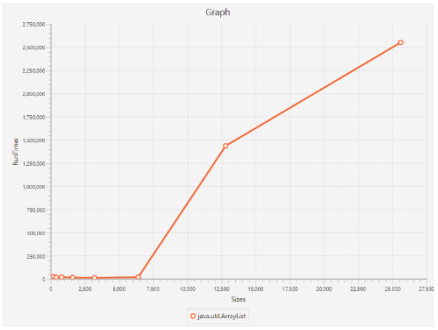
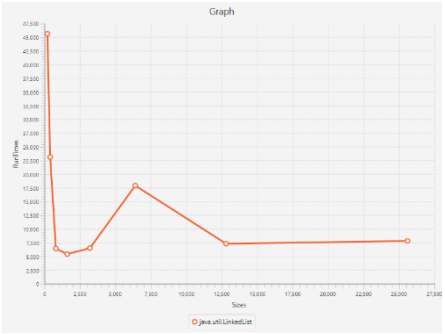
CSC 1120 Lab Seven Big-O Analysis

Benchmarking addToFront on the java.util.ArrayList Implementation



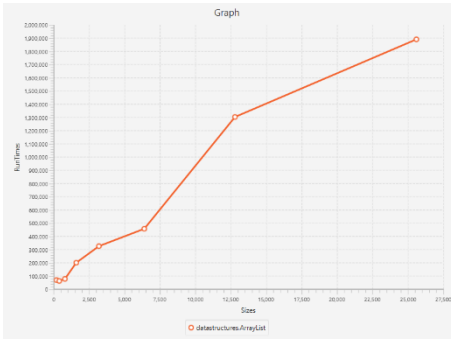
For addToFront on the java.util.ArrayList Implementation, we can say that the Big-O notation is O(n). This means that the running time of the algorithm grows linearly with the size of the array. And we can see this because the graph demonstrates a linear behavior. This is because as the size increases, the RunTimes also increases. Furthermore, we can see that the graph lines formed on the graph have constant gradients.

Benchmarking addToFront on the java.util.LinkedList Implementation



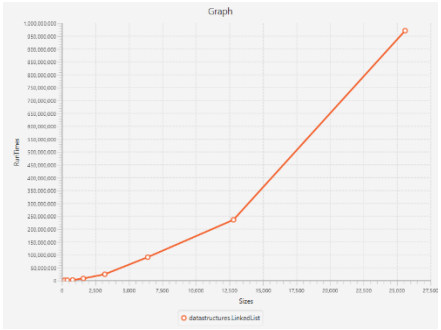
For addToFront on the java.util.LinkedList Implementation, we can say that the Big-O notation is O(1). This means that the running time of the algorithm is constant, regardless of the size of the array. And we can see this because the graph demonstrates a constant behavior. This is because as the size increases, the graph lines formed on the graph have a gradient that is very almost close to zero.

Benchmarking indexedContains on the datastructures.ArrayList Implementation



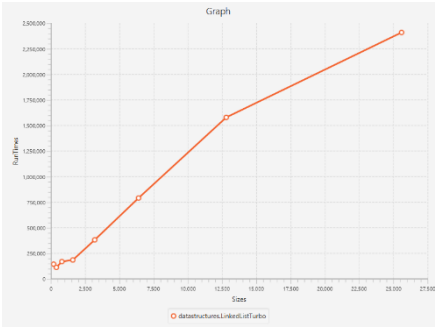
For indexedContains on the java.util.ArrayList Implementation, we can say that the Big-O notation is O(n). This means that the running time of the algorithm grows linearly with the size of the array. And we can see this because the graph demonstrates a linear behavior. This is because as the size increases, the RunTimes also increases. Furthermore, we can see that the graph lines formed on the graph have constant gradients.

Benchmarking indexedContains on the datastructures.LinkedList Implementation



For indexedContains on the java.util.LinkedList Implementation, we can say that the Big-O notation is O(n^2). This means that the running time of the algorithm grows quadratically with the size of the array. And we can see this because the graph demonstrates a quadratic behavior. This is because as the size increases, the running time of the algorithm grows by 4 times. Furthermore, we can see that the graph lines formed on the graph form half of a parabola.

Benchmarking indexedContains on the datastructures.LinkedListTurbo Implementation



For indexedContains on the java.util.LinkedListTurbo Implementation, we can say that the Big-O notation is O(n). This means that the running time of the algorithm grows linearly with the size of the array. And we can see this because the graph demonstrates a linear behavior. This is because as the size increases, the RunTimes also increases. Furthermore, we can see that the graph lines formed on the graph have constant gradients.