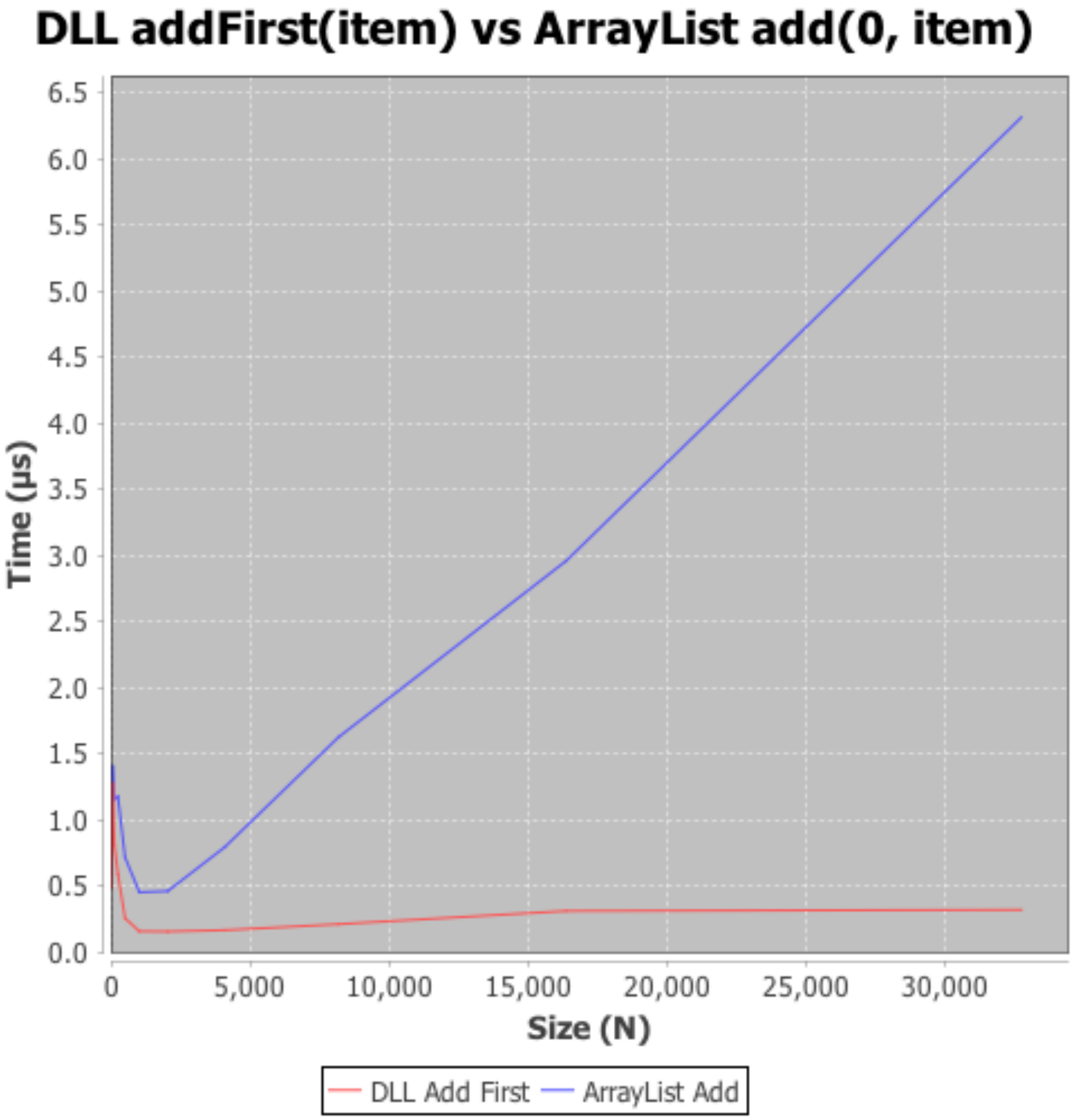
Cole Cruz
u1065058

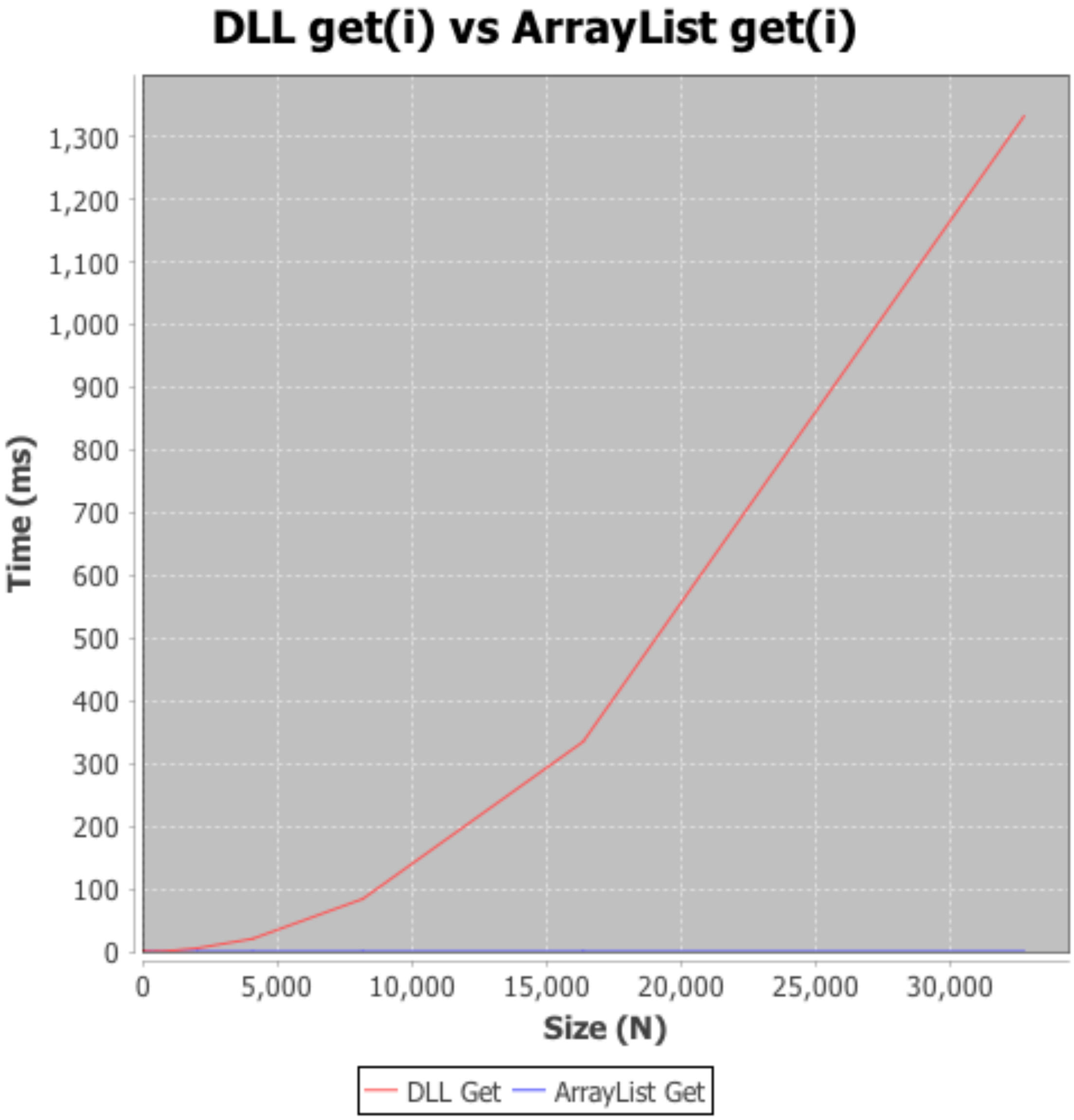
Assignment06 Analysis

1. The running time of the addFirst method for the DoublyLinkedList is O(c) as expected and performs much faster than using the add method at an index of zero for an ArrayList. The ArrayList add method looks to be O(N).



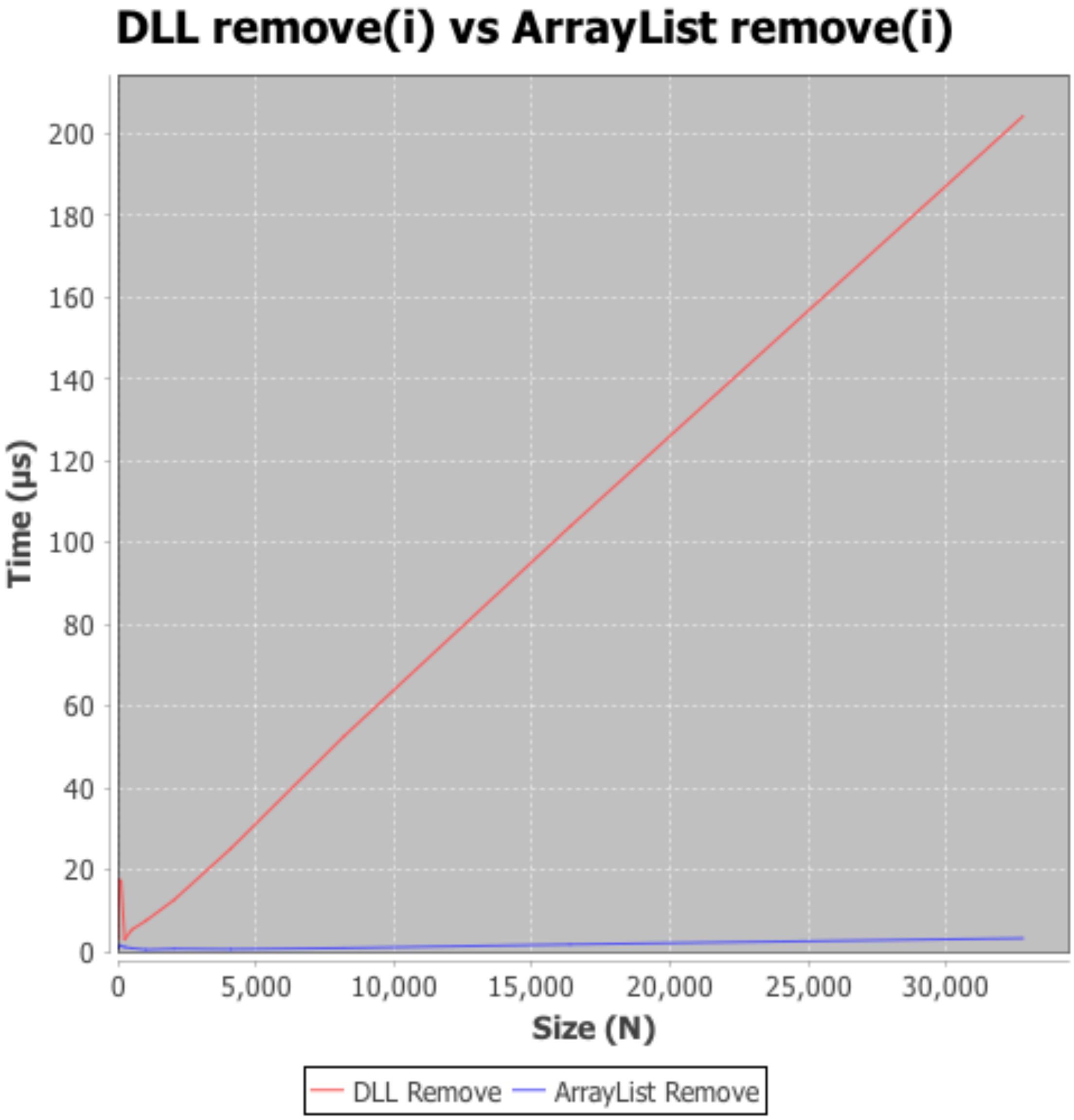
The running time of the get method for the DoublyLinkedList is about O(N) which is the expected. It performs much slower than the get method for an

ArrayList. The ArrayList get method looks to be O(c).



The running time of the remove method for the DoublyLinkedList is O(N) as expected. Although the ArrayList remove method performs much faster the

ArrayList remove method, the running time looks to be still be O(N).



2. In general, both ArrayList and DoublyLinkedList are dynamic however ArrayList is backed by an array meaning the ArrayList is limited by the size of the array and must reallocate when full. Each can hold any element including null. ArrayList also supports all list operations whereas the DoublyLinkedList does not. The methods isEmpty and size are O(c) for both

structures but the get method of ArrayList is O(c) whereas the get method for DoublyLinkedList is O(N).

3. Seeing as Java's LinkedList is a DoublyLinkedList, performance and functionality is mostly the same. Both permit all elements including null and both include the functionality of an iterator. Two differences are that Java's LinkedList implements all list operations and LinkedList also has the optimization of starting indexed methods either at the start or the end depending on what the index is closest to.
4. Either a LinkedList or an ArrayList would change the complexity of the add method from O(Nlog(N)) to O(N). For the remove function it would be most beneficial to use an ArrayList since the complexity would change from O(Nlog(N)) to O(N) and ArrayList's O(N) is better than LinkedList O(N). Assuming the get method of ArrayList performs about the same function as contains it would be best to use an ArrayList since the complexity would improve from O(Nlog(N)) to O(c).
5. I spent a total of about 7-8 hours on this assignment. After finishing the JUnit tests it was fairly easy to implement.