1,000 Elements

Method Called	ArrayList	LinkedList
Add	5 milliseconds	2 milliseconds
Sort	2 milliseconds	2 milliseconds
Shuffle	0 milliseconds	0 milliseconds
Random Get	31 milliseconds	673 milliseconds
Sequential Get	0 milliseconds	0 milliseconds

5,000 Elements

Method Called	ArrayList	LinkedList
Add	7 milliseconds	5 milliseconds
Sort	4 milliseconds	3 milliseconds
Shuffle	1 milliseconds	2 milliseconds
Random Get	23 milliseconds	9,435 milliseconds
Sequential Get	0 milliseconds	29 milliseconds

10,000 Elements

Method Called	ArrayList	LinkedList
Add	12 milliseconds	10 milliseconds
Sort	9 milliseconds	8 milliseconds
Shuffle	5 milliseconds	2 milliseconds
Random Get	26 milliseconds	23,152 milliseconds
Sequential Get	0 milliseconds	192 milliseconds

LinkedList was consistently better in terms of adding elements to the list, which is consistent with our findings in class. ArrayList would have to go through each element to find the next available space, but the LinkedList knows where the last element is, so it can simply append to the list. The sorting was about the same for both groups, as there is minimal difference in those acts between the lists.The same goes for shuffle, with the exception of of the 10,000 elements trial when ArrayList was slower. Conceptually, this makes sense as these acts are relatively similar for both types of lists. As for getting elements, ArrayList is a clear winner in this section. The RandomGet section is relatively more simple for the ArrayList because it just takes the time

to loop through the random number generation and then go to that index. For the LinkedList, in the worst case scenario, the loop may be going through all of the elements which is why it took over 23 seconds on 10,000 elements. For sequential get, ArrayList simply goes to the element in the list. This takes no time at all, because every element takes the same amount of time to get. However, for the LinkedList, each element the loop reaches, the longer it takes to get it. That is why the data is consistent with our findings in class.