Lakshya Mehta

1000 elements	Array List	Linked List
Populate	14.72 ms	4.72 ms
Sort	3.21	6.72
Shuffle	1.91	2.61
Random Get	104.38	1077.45
Sequential Get	0.04	1.41

5000 elements	Array List	Linked List
Populate	32.43 ms	23.71 ms
Sort	19.68	19.37
Shuffle	4.49	4.98
Random Get	117.17	15585.67
Sequential Get	0.33	73.80

10000 elements	Array List	Linked List
Populate	31.29 ms	15.95 ms
Sort	28.52	20.01
Shuffle	6.58	8.62
Random Get	78.62	34566.27
Sequential Get	0.63	332.39

For the populate operation, the linked list is by far better, which makes sense since you just point to a new space in memory instead of having to move all of the data to a new memory location everytime you add an item. For the sort, the linked list seems to be better but only in the larger sets, which I guess makes sense since the overhead associated with getting probably becomes less of a factor than the overhead from moving the data for a linked list. For the shuffle, the array list is faster, which if all you're doing is getting and setting to swap between two random indices, makes sense because of how easy it is to index an arraylist. Random get is much better for the arraylist than the linkedlist, and it shows how the arraylist get is a constant time, while the linked list get scales approximately linearly with the set size. For the sequential

get, there is a similar magnitude difference between the linked and array list, and follows the same reason.