Jace Halvorson
Carsten Doepnerhove

Method	Linked Complexity O()	Array Complexity O()	Explanation
boolean add(T element)	n	1 (when it has to resize it's n)	LinkedList uses one size-dependent loop for this method. ArrayList can simply add to the end with indexing, but if the list is full it has to resize, which uses one loop to copy data over to a new list.
Boolean add(int index, T element)	n	n	LinkedList uses a size-dependent loop to find the index to add the element to. ArrayList can place the the element at the index right away, but it loops through the rest of the list to push everything to the right
Void clear()	1	1	There are only 4 lines to run in this method. It has no loops so it is constant no matter how high n is.
T get(int index)	n	1	LinkedList uses a loop to iterate through until at the index. ArrayList gets it directly from an array so only one line of code is required.
Int indexOf(T element)	n	n	This method iterates through the list until it finds T element, which means it depends on the parameter, but its maximum time complexity is O(n). If the list is sorted, it stops iterating when T element > list[i].
Boolean isEmpty()	1	1	Very simple method, just checks one conditional (if length == 0). Returns the boolean value that the conditional evaluates to.

Int size()	1	1	Returns ‘length’ or ‘size’. Only has one command with no loops. Could also be called int getSize().
Void sort()	n^2	n^2	This method contains a nested size-dependent loop. That means the complexity is $O(n^2)$, which means it will be slow to process large amounts of data.
T remove(int index)	n	n	LinkedList uses a size-dependent loop to find the Node at index, ArrayList finds it right away but iterates through the rest of the list to fill in the gap
Void greaterThan(T element)	n	n	LinkedList and ArrayList both use 2 size-dependent loops but because they’re not nested loops, the complexity is O(n)
Void lessThan(T element)	n	n	LinkedList and ArrayList both use 2 size-dependent loops but because they’re not nested loops, the complexity is O(n)
Void equalTo(T element)	n	n	LinkedList and ArrayList both use 2 size-dependent loops but because they’re not nested loops, the complexity is O(n)
String toString()	n	n	Iterates through entire list to add to an output string.