Zachary Smith (smi00903) & Fahia Tabassum (tabas015)

Method	**ArrayList Runtime**	**Linked List Runtime**	**Explanation**
Boolean add(t element)	O(n)	O(1)	ArrayList has to iterate through every element of the list and has to place the new element. Our LinkedList implementation is O(n) because it loops through the entire list until the end but you can theoretically make it O(1) with a variable pointing to the end of the list.
Boolean add(int index, t element)	O(n)	O(n)	ArrayList iterates over the length of the entire array in the case of resizing (adding element) the array. LinkedList must iterate over all items before the desired index.
Void clear()	O(1)	O(1)	Both methods don't have to iterate, they will simply create new objects, similar to their constructors.
Boolean contains(T element)	O(n)	O(n)	Both methods must iterate until they find the desired element.
T get(int index)	O(1)	O(n)	ArrayList can simply return the item without iteration. LinkedList must iterate over all items before the desired index.

Int indexOf(T element)	O(n)	O(n)	Both methods must iterate until they find the desired element.
Boolean isEmpty()	O(1)	O(1)	Both methods check that at least one variable exists within them.
Int lastIndexOf(T element)	O(n)	O(n)	Both methods must iterate over all elements and run a comparison on each item.
T set(int index, T element)	O(1)	O(n)	ArrayList just simply replaces the element at the given index. LinkedList must iterate over all items before the desired index before replacing the element.
Int size()	O(1)	O(1)	Both classes keep a variable for size, so both methods simply return this variable
Void sort()	O(n^2)	O(n^2)	Both methods use a bubble sort, so the time complexity for both of them is O(n^2).
Boolean remove(T element)	O(n)	O(n)	ArrayList iterates until it finds the desired element, removes it, then continues iterating over the remaining items to put them in the correct position. LinkedList must iterate over all items before finding the desired element but when it finds it, it removes the element without any further iteration.

T remove(int index)	O(n)	O(n)	ArrayList iterates until it finds the desired index, removes it, then continues iterating over the remaining items to put them in the correct position. LinkedList must iterate over all items before finding the desired element but when it finds it, it removes the element without any further iteration.
String toString()	O(n)	O(n)	Both have O(n) time complexity as they iterate over each of the elements in their lists.