ArrayList clear and removeAll method

[ArrayList<E>](http://data-structure-learning.blogspot.com/2015/05/java-collections-part-5list-interface.html) class is a backed by an array and provides several additional features that arrays can’t. There are several [differences](http://data-structure-learning.blogspot.com/2015/05/difference-between-arraylist-and_28.html) and [similarities](http://data-structure-learning.blogspot.com/2015/05/similarities-between-arraylist-and.html) between ArrayList<E> and LinkedList<E>. Also there are [differences between ArrayList<E> and Vector<E>](http://data-structure-learning.blogspot.com/2015/05/difference-between-arraylist-and-vector.html).

Now let us understand clear() method first. After that we will compare with [removeAll(Collection<?> c)](http://data-structure-learning.blogspot.com/2015/05/removeall-operation-on-arraylist.html) method.

clear() method is used to remove all the elements from the list. It forces all the elements in backing array to be null that will let Garbage collector do its work in time.

Below is the code clear() taken from JavaDoc.

/\*\*

\* Removes all of the elements from this list. The list will

\* be empty after this call returns.

\*/

**public** **void** clear() {

modCount++;

// clear to let GC do its work

**for** (**int** i = 0; i < size; i++)

elementData[i] = **null**;

size = 0;

}

As we can see from above code the clear method iterates till the size of list and forces each entry to null.

Hence the time complexity for clear() is O(n).

We can also remove all the elements through removeAll(Collection<?> c).

Example for removing all elements

**public** **static** **void** removeAllDemo(){

List<Integer> listA = **new** ArrayList<Integer>();

listA.add(17);

listA.add(25);

listA.add(50);

listA.add(17);

listA.add(92);

listA.add(67);

listA.add(89);

listA.add(25);

listA.add(17);

System.***out***.println(listA);

listA.removeAll(listA);

System.***out***.println(listA);

}

In removeAll(..) method every element in specified collection is searched in this collection. For every element to be searched it takes O(n) time. For n elements it takes O(n^2) time.

If all the elements are to be removed from Collection then use clear() method.