Comparison of list Interface Method

Method Name	Array List [1]	Linked List [2]	Vector [3]
Add	add(E e) add(int index, E element) addAll(collectiom c) addAll(int index , collection c) set(int index, E element)	add(E e) add(int index,E element) addAll(collectiom c) addAll(int index , collection c) addFirst(E e) addLast(E e) offer(E e) offerFirst(E e) offerLast(E e) push(E e)	add(E e) add(int index, E element) addAll(collectiom c) addAll(int index , collection c) addElement(E e) set(int index, E e) setElementAt (E element, int index) insertElementAt (E element,int index)
Remove	remove(Object o) remove(int index) removeRange (int formIndex - int toIndex)	rermove() remove(Object o) remove(index) removeFirst() removeFirstOccarrence (Object o) removeLasr() removeLastOccarance (Object o) poll() pollFirst() pollLast() pop()	remove(Object o) remove(index) removeAll() removeAllElements() removeElement(Object o) removeAt(int index) removeRange(int formIndex- int toIndex)
Search	contains(Object o) get(index) indexOf(Object o) lastIndexOf(E element) isEmpty()	contains(Object o) element() get(index) getFirst() getLast() indexOf(Object o) lastIndexOf(Object o) peek() peekFirst() peekLast()	contains(Object o) containAll(collection c) elementAll(int index) equals(Object o) indexOf(Object o) isEmpty() firstElement() lastIndexOf(Object o) lastIndexOf(Object o, int index) subList(int formIndex , int toIndex)
	clear() clone ensureCapasity size() toArray() trimToSize()	clear() clone() desendingIterator() listItarator(int index) size() toArray()	capasity() clear() clone() contains() ensureCapasity() equals() hashcode() setSize(int newSize) toString() trimToSize()

Oparation	ArrayList	LinkedList	vector
add	*[4]* *O(n/2)* is average, *O(1)* best case (end of list), *O(n)* worst case (start of list)	*O(n/4)* is average, *O(1)* best case *O(n/2)* worst case(middle of list)	
remove	*O(n/2)* is average, *O(1)* best case (end of list), *O(n)* worst case (start of list)	*O(n/4)* is average, *O(1)* best case *O(n/2)* worst case(middle of list)	
search	O(1)	O(n)	
Memory	Less	More	
Synchronized	no	no	yes
Thread	no	no	yes
Data Growth Methods	50%	-	100%
Null Allow	Yes [5]	No [6]	Yes [7]

Reference:
[1] https://www.tutorialspoint.com/java/util/java_util_arraylist.htm
[2] https://ww w.tutorialspoint.com/java/util/java_util_linkedlist.htm
[3] https://www.tutorialspoint.com/java/util/java_util_vector.htm
[4] https://stackoverflow.com/questions/322715/when-to-use-linkedlist-over-arraylist
[5] httpsh://stackoverflow.com/questions/27107072/adding-null-values-to-arraylist
[6] https://stackoverflow.com/questions/38112348/why-can-we-add-null-values-to-linkedlist-__ despite-strong-discouragement-of-doing
[7] https://stackoverflow.com/questions/30908470/arraylist-vs-vector-performance-in-single- threaded-application