ArrayList Implementation

public class ArrayList {
    private Object[] intArr;
    int numElems;  // number of elements
    /* Add e to the end of this ArrayList */
    void add(Object e) { // Not checking edge cases
        numElems++;
        if (numElems == intArr.length)
            create tempArr of length intArr.length*2
            copy elements from intArr to tempArr
            set intArr = tempArr
    }

    /* Remove element at index */
    void remove (int index)

        numElems--
        Shift elements after index to the left
        if (numElems < intArr.length/2)
            create tempArr of length intArr.length/2
            copy elements from intArr to tempArr
            set intArr to tempArr

Summarize:

Running Time: ArrayList vs. (sufficiently large) array

<table>
<thead>
<tr>
<th>Operation</th>
<th>ArrayList</th>
<th>array</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>O(n)</td>
<td>O(1)</td>
</tr>
<tr>
<td>remove</td>
<td>O(n)</td>
<td>O(1) (set to -1)</td>
</tr>
<tr>
<td>get</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>indexOf</td>
<td>O(n)</td>
<td>O(n)</td>
</tr>
<tr>
<td>contains</td>
<td>O(n)</td>
<td>O(n)</td>
</tr>
<tr>
<td>size()</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
</tbody>
</table>

return numelems < 1 return array.length

in add(): numelems++
remove(): numelems--

ArrayLists also waste space!

Suppose intAir reaches capacity: [a b c ... p]

For next add, intAir length doubled: [0 1 15 16 31]

If no more inserts: [0 1 16 31]

Wasted space (Space allocated for objects, not primitive data)
Problem with arrays + Array Lists:
- For add/remove in the middle of the list, elements must be shifted.

Reason: For both, elements are stored adjacent in memory.

Instead, let's make elements non-adjacent:

But now we lost ordering.

So what additional info can we keep with each element?
⇒ Pointers to next element!

Linked List - series of nodes, not always adjacent in memory.
Requires: import java.util.LinkedList.

LinkedList<String> list = new LinkedList<String>();

How is a LinkedList implemented?
2 Classes: Node, LinkedList.

Node: (already built into Java!)

Public class Node {
    Object element;
    Node next; //something else
}

Public class LinkedList {
    Node first; //Node last...
    //something else

    /* Constructor */
    public Node (Object e)
    {
        this.element = e;
    }

    /* Constructor */
    public LinkedList ()