

Lecture 12 – ArrayLists and Runtime

Summarize Worst-Case Runtimes (in terms of number of elements in the list)

	LinkedList	MutableList (Link)	ArrayList
size	constant	constant	constant
addFirst	constant	constant	
addLast		constant	
get(index)	linear	linear	constant

Runtime of AddLast/AddFirst with Resizing

```
public class ArrList {
    String[] theArray; // the underlying array that stores the elements
    int eltcoun;       // how many elements are in the array
    int end;           // the last USED slot in the array

    private void resize(int newSize) {
        // make the new array
        String[] newArray = new String[newSize];
        // copy items from the current theArray to newArray
        for (int index = 0; index < theArray.length; index++) {
            newArray[index] = this.theArray[index];
        }
        // change this.theArray to refer to the new, larger array
        this.theArray = newArray;
    }

    public void addLast(String newItem) {
        if (this.isFull()) {
            // add capacity to the array
            this.resize(this.theArray.length * 2);
            // now that the array has room, add the item
            this.addLast(newItem);
        } else {
            if (!(this.isEmpty())) {
                this.end = this.end + 1;
            }
            this.eltcoun = this.eltcoun + 1;
            this.theArray[this.end] = newItem;
        }
    }
}
```

<pre>public class ArrTest { ArrList flavors = new ArrList(2); flavors.addLast("mint") flavors.addLast("grape") new Course("cs1410", 200) flavors.addLast("lemon") flavors.addLast("cherry") } ----- - environment flavors → @1221</pre>	@1221	ArrList theArray: @1222 end: 1 eltcoun: 2
	@1222	"mint"
	@1223	"grape"
	@1224	Course("cs1410", 200)
	@1225	
	@1226	
	@1227	
	@1228	

How many resizes get done across N calls to addLast? How does this affect runtime?

```
ArrayList flavors = new ArrayList(2);
```

	Resize by 1	Resize by 2	Resize by double
flavors.addLast("mint")			
flavors.addLast("grape")			
flavors.addLast("lemon")			
flavors.addLast("cherry")			
flavors.addLast("mango")			
flavors.addLast("orange")			
flavors.addLast("coffee")			

Enabling AddFirst – Leave space at front and end of the array to avoid resizing

```
public class ArrTest1 {  
    ArrayList flavors1 = new ArrayList(6);  
    flavors.addLast("mint")  
    flavors.addLast("grape")  
    flavors.addLast("lemon")  
    flavors.addLast("cherry")  
}
```

```
public class ArrTest2 {  
    ArrayList flavors2 = new ArrayList(6);  
    flavors.addLast("berry")  
    flavors.addFirst("kiwi")  
    flavors.addFirst("lime")  
    flavors.addLast("apple")  
}
```

flavors1

ArrayList theArray: @XXXX end: 3 eltcount: 4
"mint"
"grape"
"lemon"
"cherry"

flavors2

ArrayList theArray: @XXXX end: eltcount:
"berry"

flavors2

ArrayList theArray: @XXXX end: start: eltcount:

flavors2

ArrayList theArray: @XXXX end: start: eltcount:

flavors2

ArrayList theArray: @XXXX end: start: 3 eltcount:

flavors1 (again)

ArrayList theArray: @XXXX end: start: 3 eltcount:

chocolate	end
mint	start
grape	
lemon	
cherry	

addLast(chocolate)