Linked Structures

You’ll need access to a laptop.

(A $O(n)$ algorithm with a large constant factor!)
Appending

Snarf Sep24InClass

Read ExpandingArray, and then:

http://goo.gl/GzP8g
Appending

Snarf Sep24InClass

Read it, and then: http://goo.gl/GzP8g

and then: http://goo.gl/S7UaJ
We have a problem...

ExpandingArray

.get(): $O(1)$  You couldn’t hope for better!

.add(): $O(n)$  Which means $O(n^2)$ for n operations...

Vitally important: ArrayLists have better Big-O performance than this!
Vitally important: ArrayLists have better Big-O performance than this!
Vitally important: ArrayLists have better Big-O performance than this!

Duke Comp. Sci. is great!
Vitally important: ArrayLists have better Big-O performance than this!

Duke

Comp.  Sci.  is

great!
Vitally important: ArrayLists have better Big-O performance than this!

Duke Comp.

Sci. is great!
Block Structures

Duke  Comp.  Sci.

is

great!

Vitally important: ArrayLists have better Big-O performance than this!
Vitally important: ArrayLists have better Big-O performance than this!

Duke Comp. Sci. is great!
Vitally important: ArrayLists have better Big-O performance than this!

Duke Comp. Sci. is great!
| Duke | Comp. | Sci. | is | great! |

**Vitally important:** ArrayLists have better Big-O performance than this!
Each element has a pointer to the next one!

Linked Structures

front

Duke

Comp.

Sci.

is

back

great!
Linked Structures

Duke → Comp. → Sci. → is → great!
Linked Structures

http://goo.gl/mDiBQ
...do we have a solution?

Expanding Array
.
.get(): $O(1)$  

You couldn’t hope for better!
.
.add(): $O(n)$  

Which means $O(n^2)$ for n operations...

Linked List
.
.get():
.
.add(): $O(1)$  

Best it can be!

Super important: Java has a LinkedList class. It’s doing something slightly different than this one, but the Big-O is the same.
Linked Structures

How many operations does it take to get this element?

Duke → Comp. → Sci. → is → great!

front → back
Linked Structures

Duke → Comp. → Sci. → is → great!

front

back

iterator
Linked Structures

front

Duke → Comp. → Sci. → is → great!

back

iterator
Linked Structures

Duke → Comp. → Sci. → is → great!

front

back

iterator
Linked Structures

Duke \rightarrow Comp. \rightarrow Sci. \rightarrow is \rightarrow great!

front

back

iterator
Linked Structures

Duke → Comp. → Sci. → is → great!

http://goo.gl/sTKFr
...do we have a solution?

ExpandingArray
.get(): $O(1)$  
.add(): $O(n)$

You couldn’t hope for better!

Which means $O(n^2)$ for n operations...

Linked List
.get(): $O(n)$
.add(): $O(1)$

Which means $O(n^2)$ for n operations...

Best it can be!
What about...

Duke  Comp.  Sci.  is  great!

...adding at the front?

Or in the middle?
Science!

http://goo.gl/nbF0j

(and the usual survey)

http://goo.gl/Z09dK