Linked List Practice

As you arrive, please snarf the files for class today “recitation_8_classwork”. In that code, you will find my solution to the prep work. If you had trouble with removeLongestString, take a look at mine and see if that helps.

If you didn’t have problems with removeLongestString, go ahead and start implementing the compareTo method for StringLinkedList.

If you didn’t get your exam Wednesday, you can get it from me after class.

Today in class: Practice Tricky Linked Lists Problems

• You will probably never write a linked list from scratch in your life
• BUT...Linked List is by far the simplest of the linked data structures; if you can’t do this with ease, you’ll struggle as we move on to things like trees and graphs

Any questions with the prep work?

• You’ve just snarfed my solution

Question 1: DoubleList

• Takes a list and doubles each element [a,b,c] -> [a,a,b,b,c,c]
• When you finish, make sure that your code passes the testDouble unit test
• Feel free to work on compareTo if you finish early (we probably won’t actually cover it in class)
Question 2: Reverse

- \([a,b,c]\) -> \([c,b,a]\)
- Requires a bit of care to get right

Question 3: moveToEnd

- Moves \(n\) elements from the beginning of the list to the end. 
\([a,b,c,d,e]\) -> moveToEnd(2) -> \([c,d,e,a,b]\)
- Your code should run in \(O(n)\) time, meaning you shouldn’t be calling \(\text{addAtEnd()}\). Hey, BTW, what if you did call \(\text{addAtEnd}()\) – what would the \(O\) be, where \(n\) = list length?
- You don’t need to consider invalid cases (e.g. what happens if you try to move more than the length)
- You should consider edge classes:
  - If you ask it to move zero elements to the end (leaves list unchanged)
  - If you ask to move exactly length elements (leaves list unchanged)
- When you are finished, submit via Ambient