W3D2 Lab 2: LinkedList

Description:

In this lab you will add a .sort() and a .contains() method to the LinkedList class. Again the .contains() method will assume that the list is already sorted and then does a binary search through its items.

Provided:

We've provided the LinkedList class, and you can use the sort code from ArrayList (provided for the previous lab) and also the binary search code that was provided with the previous lab.

Instructions:

Start by adding the .sort(comparitor) method to LinkedList. Notice that because we've use the same names for all the methods in ArrayList and LinkedList copy/pasting the method makes it work right away. This is the power of what is called "Program to Interface" also known as P2I.

Also notice that when swapping items we do not actually swap the nodes, instead we are just swapping the elements that are stored in those nodes.

Be sure to write Mocha / Chai tests to check if your .sort() method is working.

Once you have .sort() working you can also add .contains(element). If you wrote a nice version of contains in your ArrayList you should be able to just copy and paste it over as well. Be sure to also write tests for it.

Last Step:

Now although Copy and Pasting the .contains() from ArrayList works (or is easy to make work), it does not actually make efficient use of the LinkedList's nature. It uses numeric indexes into the list, which are always slow!

Update your .contains() method to use references to nodes instead of numeric indexes (the begin, end and middle variables). To be specific if the item found in the middle is too big, make your begin reference point to the middle node. Otherwise make the end reference point to the middle node.

Because we no longer have index values we can also no longer return -1 for not found and the index value for when it is found. Instead just return true when it's found and false when not.

The test code you wrote for the copy/pasted version will not need to be updated to test this more efficient way of working a linked list.

Even with these updates, is such a binary search on a LinkedList more efficient than a linear search?