CS 0445 Spring 2024
Recitation Exercise 5

Introduction:

We have discussed recursion in lecture (see Lectures 11 and 12) and you will be using recursion in your next programming project. To help to understand recursion, in this exercise you will implement a very simple linked list class utilizing exclusively recursive methods.

Details:

You should be familiar with linked lists and in particular with a circular, doubly-linked list (you are using this type of list in Assignment 2). In this exercise you will complete a very simple doubly linked list class, called DoubleRecList, in which **all of the methods are implemented recursively**. This class has only 3 public methods: 1) a constructor to initialize the list from an array of String, 2) the toString() method and 3) a reverse() method. The first two methods have been provided for you. Look at these methods and the comments carefully. Run the main program (CS445Rec5.java) as it is given to trace the output.

Once you are comfortable with the DoubleRecList class and the methods provided, you will add one additional method – the reverse() method. This method will do as it states – reverse the data in the list. It should do this by reversing the Nodes in the list and **without creating any new Nodes** and without using any auxiliary data structures (ex: like an array or ArrayList). This method may be a bit challenging to conceive at first but after your TA traces the idea with you in recitation and after a bit of pencil and paper work yourselves you should be able to figure this out. Figuring out this method will go a long way toward your understanding recursive programming in general and will also help you with Assignment 3 (once it is posted).

Once you have completed the reverse() method remove the comments from the CS445Rec5.java program and compile and run it again. Verify that your data is in fact being reversed.

Note: To run the program you will need the following files:
> CS445Rec5.java (main program)
> DoubleRecList.java (file that you must modify)
> Node.java (external Node class – make sure to use this version)

Hints:

As with the other recursive methods in the DoubleRecList class, you should write a private recursive method to do the actual work.

Since the list is doubly-linked, one way to approach this is to move BACKWARD through the original list, extracting each node and adding it to the END of a new list. Alternatively, you can move FORWARD through the original list, extracting each node and adding it to FRONT of a new list. In either case you are disconnecting and reconnecting existing nodes – without making new nodes in your list. You can do either of these approaches using two reference variables in your recursive calls – one to traverse the old list and one to build the new list. Be careful about detecting the end of the list (since it is circular).