CIS351

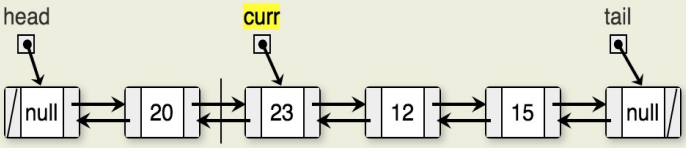
Homework 7

Submission Instructions

1. Submit completed **DoubleList.java** through in Blackboard.
2. DO NOT forget to mention your full name in the Java class documentation.
3. submit a filled in **cover sheet** in Blackboard.

Introduction

The singly linked list allows for direct access from a list node only to the next node in the list. A doubly linked list allows convenient access from a list node to the next node and also to the preceding node on the list. The doubly linked list node accomplishes this in the obvious way by storing two pointers: one to the node following it (as in the singly linked list), and a second pointer to the node preceding it.



The most common reason to use a doubly linked list is because it is easier to implement than a singly linked list. While the code for the doubly linked implementation is a little longer than for the singly linked version, it tends to be a bit more "obvious" in its intention, and so easier to implement and debug. Whether a list implementation is doubly or singly linked should be hidden from the List class user.

Like our singly linked list implementation, the doubly linked list implementation makes use of a header node. We also add a tailer node to the end of the list. The tailer is similar to the header, in

that it is a node that contains no value, and it always exists. When the doubly linked list is initialized, the header and tailer nodes are created. Data member head points to the header node, and tail points to the tailer node. The purpose of these nodes is to simplify the insert, append, and remove methods by eliminating all need for special-case code when the list is empty, or when we insert at the head or tail of the list.

Download Materials

For this homework, you need to download the following starter code from Blackboard:

- DoubleList.java
- Link.java
- DoubleListDriver.java

Part 1 - Doubly Linked List Iterator

Complete the unfinished iterator methods in DoubleList.java so that they satisfy the requirements of the Iterator Interface.

Part 2 - Additional List Methods

Implement the following methods in DoubleList.java class:

- add(int index, E item)
- remove(int index)
- reverse()

Grading Criteria

Total points: 100 points

Doubly Linked List Iterator - 85 pts
Additional List Methods - 15 pts

Don't forget
the submission process or
the grading criteria or
the cover sheet.

Farzana Rahman / frahman@syr.edu