

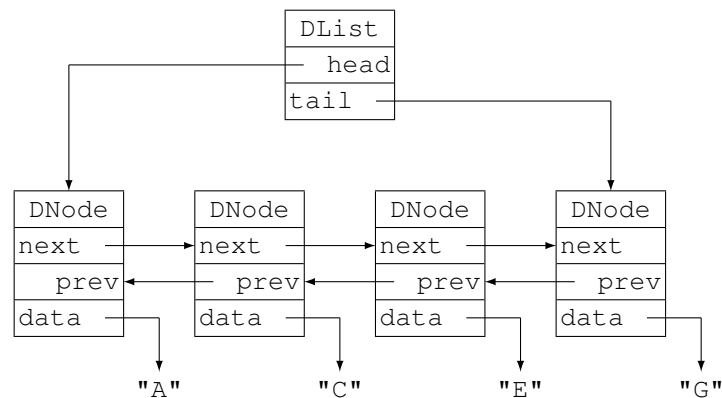
CMPSCI 187, Spring 2015 Discussion #9

Doubly-Linked Lists: Individual Handout

Marc Liberatore and John Ridgway

30 March 2015

Removing and inserting elements into a linked-list in a particular location, rather than at the beginning or the end, can be a somewhat painful operation, as you have to find the node *before* the one you want to remove, or after which you want to insert. One of the ways to make this easier is to add an extra link to the nodes of the list, a *previous* or *back* pointer, which refers to the node before this one in the list. Such a doubly-linked list might resemble the following:



Here is code that will add an element after a particular node in a doubly-linked list. If `insertAfter` is `null` it means to insert the element at the beginning of the list.

```
1 public void addAfter(T element, DNode<T> insertAfter) {
2     DNode<T> insertBefore;
3     if (insertAfter == null) { // add at start of list
4         insertBefore = head;
5     } else {
6         insertBefore = insertAfter.getNext();
7     }
8     DNode<T> newNode = new DNode<T>(element, insertBefore, insertAfter);
9     if (newNode.getNext() == null) {
10        tail = newNode;
11    } else {
12        newNode.getNext().setPrevious(newNode);
13    }
14    if (newNode.getPrev() == null) {
15        head = newNode;
16    } else {
17        newNode.getPrev().setNext(newNode);
18    }
19 }
```

Question 1: To satisfy yourself that this method always works draw a picture that shows adding a node to an empty list; a second that shows adding a node to the beginning of a non-empty list; and a third that shows adding an element in the middle of a list.

Question 2: Write a method `void removeNode (DNode<T> nodeToRemove)` that takes a reference to a particular node in the list and removes that node.

Question 3: Write a method `moveInto` with the following signature:

`void moveInto (DList<T> other, DNode<T> insertAfter)`

The method moves all of the elements from `other` into the current list, after the element `insertAfter`. If `insertAfter` is `null`, they should be inserted at the start of the list. Note that the `other` list should end up empty.