STUDENT OUTLINE

Lesson 31 – Doubly-Linked Lists

INTRODUCTION: This lesson presents a variation of singly-linked lists, a doubly-linked list. There are occasions when it might be more efficient to have nodes with pointers going both ways - to the right and to the left. Working with such lists involves twice as many internal pointers, as the diagrams will illustrate.

The key topic for this lesson is:

A. Doubly-Linked Lists
B. Deleting from a Doubly-Linked List

VOCABULARY: DOUBLY-LINKED LISTS

DISCUSSION:

A. Doubly-Linked Lists

1. The node of a doubly-linked list will contain the information field(s) and two reference fields. One reference will refer to a previous node while the other reference will refer to the next node in the list.

2. The following class definitions will be used in this student outline.

```
public class DListNode
{
  private Object value;
  private DListNode next;
  private DListNode previous;

  // Constructor:
  public DListNode(Object initValue,
                   DListNode initNext,
                   DListNode initPrevious)
  {
    value = initValue;
    next = initNext;
    previous = initPrevious;
  }

  public Object getValue()
  {
    return value;
  }

  public DListNode getNext()
  {
    return next;
  }

  public DListNode getPrevious()
  {
    return previous;
  }
```

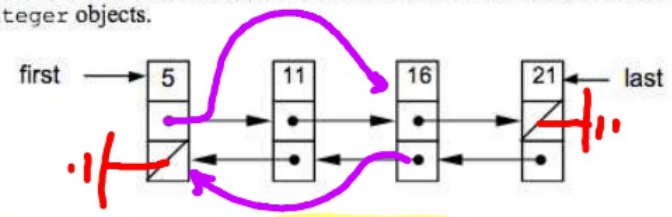
```
  public void setValue(Object theNewValue)
  {
    value = theNewValue;
  }

  public void setNext(DListNode theNewNext)
  {
    next = theNewNext;
  }

  public void setPrevioust(DListNode theNewPrevious)
  {
    previous = theNewPrevious;
  }
}
```

(B) NEXT PAGE

3. Here is a picture of a doubly linked list, of type `DListNode` containing `Integer` objects.



2 STOP SIGNS

4. A `null` value must be placed at each end of the list to signify the end of the data structure. In the diagram, a **`null`** is indicated with the diagonal line.

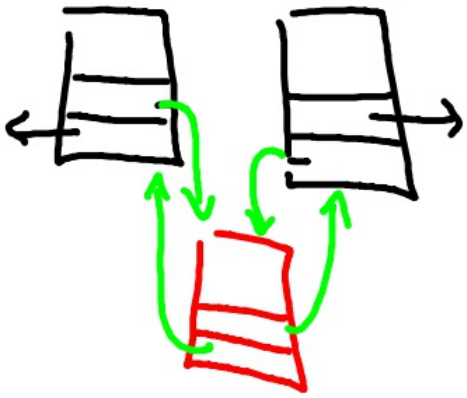
5. A doubly-linked list should have two external references to access the data structure. In the case above, `first` and `last` are the two entry points.

6. A doubly-linked list can be traversed in either direction.

4 CASES

7. Inserting values into an ordered doubly-linked list is a similar process to the algorithm used with a singly-linked list. However, the number of reference manipulations will double.

8. The addition of a new node to a position between two existing nodes will require four reference hookups.



ORDER??

B. Deleting from a Doubly-Linked List

1. There are some special cases to be aware of when deleting nodes from a doubly-linked list. The diagram in section A.3 will be used for the illustration.

2. Deleting the only node in a one-node list means that one or both of the external references are set to `null`.

3. Deleting the first node would require `first` to be changed.

4. Deleting the last node would require `last` to be changed.

5. Deleting a node between two others will require two reference manipulations.

SHOWN ON DIAGRAM PG 2

SUMMARY/ REVIEW: This data structure, a doubly-linked list, is an extension of the basic linear-linked list. The lab work will provide good practice in working with pointers and linked lists.

ASSIGNMENT: Lab Exercise L.A.31.1, *Double*