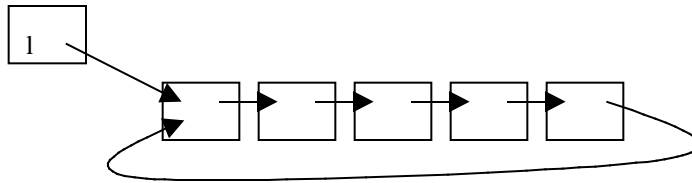


Jan 28

Circular Linked Lists

A circular linked list is one in which the last node points to the first node.



Note: if `currentNode.next == 1`, we are at the “last” node. I.e. if the list reference is the same as the `current.next` reference, then we are at the ‘last’ node of the list.

Printing all the nodes in a circular linked list

```

Static void printNode(ListNode list) {
    ListNode currentNode = list;

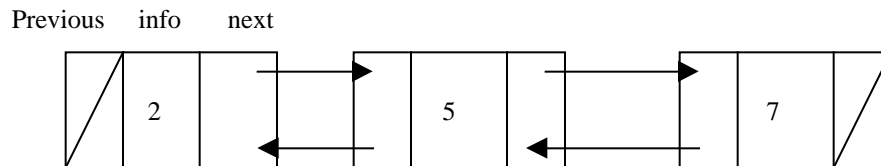
    If (list == null) {
        Return;
    }

    while (currentNode.next != list) {
        System.out.println(currentNode.info);
        currentNode = currentNode.next;
    }
    System.out.println(currentNode.info);
}
    
```

Applications of Circular Linked Lists

- polygon clipping
- round robin situations
- when you need to rotate the list, rather than the reference.

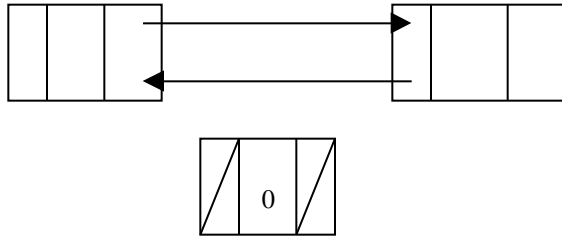
Doubly Linked Lists



Inserting into a Doubly Linked List

```

currentNode = ... // whatever current node is
ListNode newNode = new ListNode();
    
```



```

newNode.info = 1; // setting the value of newNode
newNode.next = currentNode;
newNode.previous = currentNode.previous;
currentNode.previous.next = newNode;
currentNode.previous = newNode;

```

