
Circular Linked Lists

CMPSCI 187 Discussion

21 February 2007

Name: _____

A circular linked list allows you to traverse the list indefinitely (calling `next()` from the tail will give the head of the list). Let's say a list package has the following implementation for node.

```
public class Node {
    private Object data;
    private Node next;

    public Node( Object d ) {
        data = d;
        next = null;
    }

    public Node( Object d, Node n ) {
        data = d;
        next = n;
    }

    public void setData( Object d ) { data = d; }
    public Object getData() { return data; }
    public void setNext( Node n ) { next = n; }
    public Node getNext() { return next; }
}
```

1 Assume an implementation of a circular linked list has a reference to the first and last elements of the list (member variables `Node head` and `Node tail`). Complete the following insertion member method for a circular linked list. Don't forget to handle special cases where the list is empty or there is only one element.

```
public void insertTail( Object item ) {

}

}
```

- 2** Now assume that there is only a reference to the first node in the list (there is no member variable `tail`).

```
public void insertAtTail( Object item ) {
```

```
}
```

- 3** Analyze the time complexity of each of your methods.