

CS 211 Summer 2007, Review problem set #3

1. Determine whether each of the following statements is true. If it is, provide a witness pair to prove the big-O bound.
 - (a) $30n^3 + 20n^2 + 5 = O(n^3)$
True, $c = 50, N = 3$. There are other possible answers.
 - (b) $2^{n+1} = O(2^n)$
True, $c = 2, N = 0$. There are other possible answers.
 - (c) $3^n = O(2^n)$
False.
 - (d) $\log_2 n = O(\log_3 n)$
True, $c = (\log_3 2)^{-1}, N = 1$. There are other possible answers.
2. Write a simple linked-list class called `LList`. The implementation should use generics, so that an `LList` object can be created to hold data of any arbitrary reference type. The class should have two public methods: `insertFront()` and `removeFront()`.
See next problem for solution.
3. Write an iterator class for `LList` called `LListIterator`. Implement it as an inner class in `LList`. Then show how to implement the iterator as an anonymous class.

```
// This version implements the iterator in an inner class.
```

```
import java.util.*;
```

```
class LList<T> implements Iterable {
```

```
    class LListCell<T> {
```

```
        T datum;
```

```
        LListCell<T> next;
```

```
        LListCell(T datum, LListCell<T> next) { this.datum = datum; this.next = next; }
```

```
    }
```

```
    LListCell<T> head;
```

```
    public void insertFront(T datum) {
```

```
        head = new LListCell<T>(datum, head);
```

```
    }
```

```
    public void removeFront() {
```

```
        if(head != null)
```

```
            head = head.next;
```

```
    }
```

```

public Iterator<T> iterator() { return new LListIterator<T>(this); }

class LListIterator<T> extends Iterator {
    LList<T> list;
    LListCell<T> cursor;

    LListIterator(LList<T> list) { this.list = list; cursor = list.head; }
    public T next() { T d = cursor.datum; cursor = cursor.next; return d; }
    public boolean hasNext() { return cursor != null; }
    public void remove() { }
}
}

```

4. Give closed-form solutions to the following recurrences, and prove them using induction.

(a)

$$T(0) = 5$$

$$T(n) = 3T(n-1) + 7 \text{ for } n \geq 1$$

$$T(n) = 5 \cdot 3^n + \frac{7}{2}(3^n - 1)$$

Then prove by induction.

(b)

$$T(1) = 1$$

$$T(n) = 4T(n/2) + 3n \log_2 n \text{ for } n \geq 1$$

This problem was harder than I thought; don't worry if you can't figure out the closed-form expression.

$$T(n) = 7n^2 - 3n(\log_2 n + 2)$$

Then prove by induction.

5. Compute the running time of each of the following loops.

(a)

```
int r=0;
for(int i=0; i<n; i++)
    for(int j=n; j>=0; j--)
        r++;
```

$$O(n^2)$$

(b)

```
int r=0;
for(int i=0; i<n; i++)
    for(int j=0; j<i*i; j++)
```

```
    r++;
```

$O(n^3)$

```
(c) int r=0;
    for(int i=0; i<n; i++)
        for(int j=i; j<=i; j++)
            for(int k=j; k<=n; k++)
                r++;
```

$O(n^2)$