COP 3530: Fall 2016: MidTerm Exam Review

1. Complete the following: $f(n) = O(g(n))$ means that "...

2. Indicate for each pair of expressions (A, B) in the table below, whether A is O, o, Ω, ω, or Θ of B.

	A	B	O	o	Ω	ω	Θ
a.	$3n^2 + 20n + 160$	$n^3 - 3n^2 - 20n - 160$					
b.	n^2	$n \log n$					
c.	$\log n$	$\sqrt{n}$					
d.	$\log \log n$	$\log n$					
e.	$\log^2 n$	$\sqrt{n}$					
f.	n^3	2^n					
g.	$2^{\log n}$	$\log(2^n)$					
h.	$n!$	2^n					
i.	3^n	2^n					
j.	$n^2 2^n$	3^n					

3. You are given a standard linked list class called `MyDoublyLinkedList` with the standard methods (`add, get, set, remove,` etc.. As discussed in class, the linked list class has two fields called `beginMarker` and `endMarker` pointing to the two dummy nodes at either end of the list. Also, each `Node` in the linked list has the fields `data`, `prev` and `next`.

   (a) Implement a private method `moveToFront(int idx)` that moves the node at location idx to the start of the list. It should return `void`. You may assume that there are dummy nodes at the start and at the end of the linked list. Therfore the `prev` and `next` pointers in nodex `x` and `y` can be assumed to be not null.

   (b) Time Complexity of `moveToFront` is

4. [15] You are given a `BinarySearchTree<AnyType>` class with the standard methods as shown below. Each `Node` in the linked list has the standard fields `element`, `left` and `right`. You may assume that there are no duplicates in the tree.

```
void insert( x ) --> Insert x
void remove( x ) --> Remove x
boolean contains( x ) --> Return true if x is present
boolean remove( x ) --> Return true if x was present
boolean isEmpty( ) --> Return true if empty; else false
void makeEmpty( ) --> Remove all items
void printTree( ) --> Print tree in sorted order
```

(a) Implement a method that `paste`s together two binary search trees. Assume you have two binary search trees `A` and `B` and that all values stored in `A` are strictly smaller than the values in `B`. You call call this function as follows: `A.paste(B)`. The result should be that all values in tree `B` should become part of tree `A` after the operation. Nodes of `B` are incorporated into tree `A` and tree `B` is simply made empty.

```
public void paste(BinarySearchTree<AnyType> B)
```

(b) Time Complexity of `paste()` is

5. Explain briefly the algorithm to check whether an expression consisting of parentheses is balanced.
6. Write down the time complexities of all public methods for the data structures we have discussed in class.