COSC 2P03 MIDTERM TEST
26TH OCTOBER, 2009

Time: 12:30 p.m. – 1:45 p.m. Total marks: 50 Total pages: 3
This is a *closed-book* test: no additional materials (including calculators) are permitted
Answer all questions *in your exam booklet*.

Question 1 (14 marks) – Complexity and Recursion

a) [2 marks] Express the complexity of the following code fragment using big-O notation. **You must explain how you arrived at your answer.**

```
for(i = n/2; i < n; i=i+2)
  for(j = n; j >= 1; j=j-1)
     a[i] = b[i]+j;
```

b) [4 marks] Express the complexity of the following method using big-O notation. **You must explain how you arrived at your answer.** What value is returned by the call `fred(1,4)`?

```
int fred(int x, int y)
{
  if(y == 0)
    return x;
  else if((y % 2) == 0)
    return fred(2 * x, y / 2);
  else
    return fred(x, y / 2);
}
```

c) [2x3 marks] Explain the two **fundamental** rules of recursion. For each of these rules, identify where they are applied in the recursive method from part (b) above (i.e. at which line of the method).

d) [2x1 marks] You have decided to implement a priority queue using a sorted linked list. In this priority queue, the items with the smallest key value are those that should be removed first. What are the complexities of (i) removeMin and (ii) enqueue for this implementation? No explanation is required.

Question 2 (19 marks) – Trees and Traversals

a) [2 marks] If it is possible, draw a **complete** binary tree with 8 nodes. If it is not possible, then explain why.

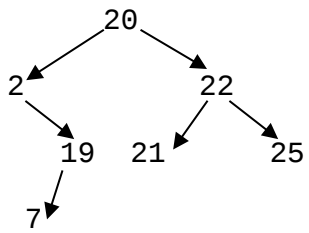
b) [2 marks] If it is possible, draw a **full** binary tree with 8 nodes. If it is not possible, then explain why.

c) [5 marks] Write a **recursive** method

```
BinaryNode findMin(BinaryNode T)
```

that will return the node containing the smallest value in the binary search tree with root T. Note: your method **must be recursive**, or no marks will be given.

d) Consider the following binary search tree:



(i) [1 mark] Indicate any pair of nodes that are **adjacent** in this tree. No explanation is required.

(ii) [1 mark] What is the **depth** of this tree? No explanation is required.

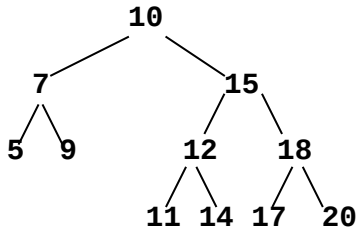
(iii) [4 marks] Draw the above tree threaded for **inorder** traversal. Show both predecessor and successor threads. Use arrows with dashed lines to represent threads.

(iv) [2 marks] Give the **preorder** traversal of this tree.

(v) [2 marks] Give the **postorder** traversal of this tree.

Question 3 (8 marks) – AVL Trees

a) [4 marks] Consider the following AVL tree:



Draw the resulting AVL tree after the insertion of a node containing the key value 13.

b) [4 marks] Given the tree **resulting from part (a) above**, draw the AVL tree after the insertion of a node containing the key value 3.

Question 4 (9 marks) – B Trees

a) [5 marks] Suppose we wish to create a B tree on a computer with a block size of 512 bytes and pointers of 8 bytes. The records we wish to store in the tree are 50 bytes each, **including** keys of 12 bytes.

(i) What is the *order* of the B tree?

(ii) What are the *minimum* and *maximum* numbers of records that can be stored in a leaf node that is not the root?

(iii) What is the *minimum* number of children for a root that is not a leaf node?

b) [2 marks] Deletion for B trees is generally implemented in a “lazy” manner. Explain what this is. Briefly explain the reasoning behind using this type of implementation.

c) [2 marks] Explain why Big-O analysis is generally not meaningful for B trees. To measure efficiency, what should we count instead of using Big-O notation?