NAME:_____ STUDENT NUMBER:_____

COSC 2P03 MIDTERM TEST
25TH OCTOBER, 2018

Time: 9:00 a.m. – 10:15 a.m.
Total marks: 50

No electronic or written materials (including calculators) are permitted.
Answer all questions *directly on the question paper*.
Write concise but complete answers.

Question	Value	Mark
1	10	
2	18	
3	9	
4	13	
Bonus	1	
Total:	50	

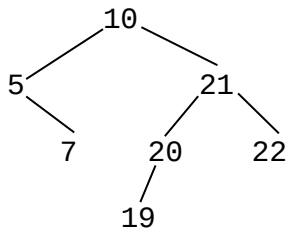
Question 1 (10 marks) – Complexity and Recursion

a) [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 m, int n)
{
  if(n <= 0)
    return m;
  else if((n % 2) == 0)
    return fred(2 * m, n / 2);
  else
    return fred(m, n / 2);
}
```

b) [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 (a) above, i.e. at which lines of the method.

Question 2 (18 marks) – Trees and Traversals
Consider the following binary search tree:



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

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

c) [2 marks] Draw the tree after the deletion of the node with value 5.

d) [2 marks] **Given the tree resulting from part (c) above**, draw the tree after the deletion of the node with value 10.

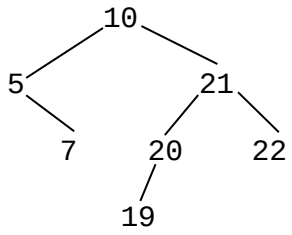
Question 2 (continued) – Trees and Traversals

e) [5 marks] Write a recursive method

```
increasingOrder(BinarySearchNode T)
```

that will traverse the **binary search tree** with root T and print the nodes in increasing order of their key values. You can make use of the method `T.printNode()` to "print" node T. Your method **must be recursive**, or no marks will be given.

f) [4 marks] Thread the following tree for inorder traversal. You should use arrows with dashed lines to represent threads, and can draw them directly on the following tree.



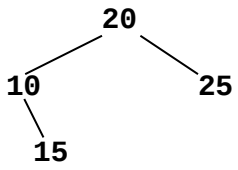
g) [3 marks] Write a method

```
ThreadedNode findMax(ThreadedNode T)
```

that will return the node containing the largest value in the threaded binary search tree with root T.

Question 3 (9 marks) – Height-balanced Trees

a) [2 marks] Consider the following AVL tree:



Draw the resulting AVL tree after the insertion of a node containing the key value 17.

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 19.

c) [3 marks] *Briefly* explain the main motivation for maintaining a height-balanced tree.

Question 4 (13 marks) – B Trees

a) [6 marks] Suppose we wish to create a B tree on a computer with a block size of 2048 bytes and pointers of 8 bytes. The records we wish to store in the tree are 200 bytes each, *including* keys of 12 bytes.

(i) [3 marks] What is the *maximum* number of children for an **index** node that is not the root? What is the *minimum* number of children for an **index** node that is not the root?

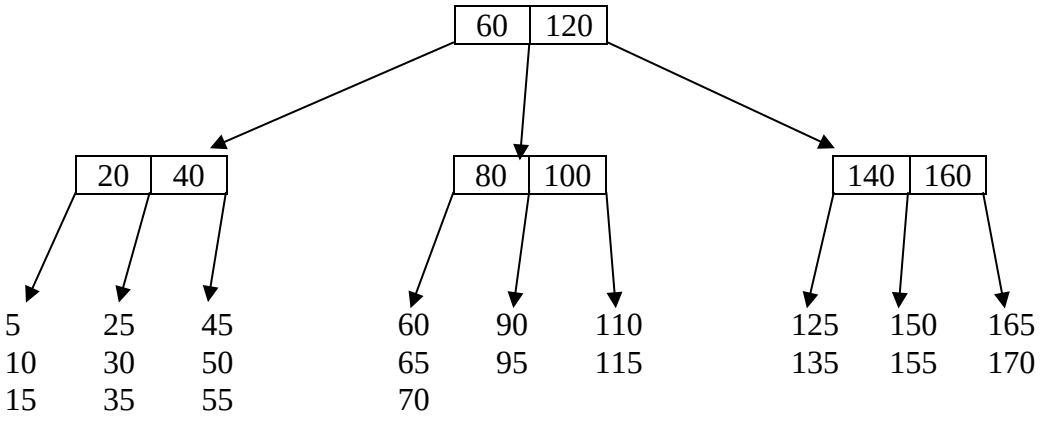
(ii) [2 marks] What are the *minimum* and *maximum* numbers of records that can be stored in a **leaf** node that is not the root?

(iii) [1 mark] What is the *minimum* number of children for a **root** that is not a leaf node?

b) [3 marks] Briefly explain why Big-O analysis is generally not meaningful for B trees. To measure efficiency for B trees, what should we count instead of using Big-O notation?

Question 4, continued – B Trees

c) [4 marks] Consider the following B-tree which has order M=3 and in which each leaf node can hold a maximum of L=3 records. Only the keys of the records are shown. Suppose that a record with key value 1 is inserted into this tree. Draw the resulting tree.



Bonus Question (1 mark): What is shown in the background of the course webpage?