THE UNIVERSITY OF TRINIDAD & TOBAGO

FINAL ASSESSMENT/EXAMINATIONS APRIL/MAY 2014

Course Code and Title: DSAL 2001 Data Stuctures & Algorithms

Programme: B.A.Sc. ICT Engineering

Date and Time: Thursday 17th April,2014 9:00A.M – 12:00 Noon Duration: 3 Hours

PLEASE READ ALL INSTRUCTIONS CAREFULLY BEFORE YOU BEGIN THIS EXAMINATION

Instructions to Candidates

1. **This paper has 4 pages and 10 questions in 3 sections – A, B and C.**
2. **You are required to answer All questions in Each Section.**
3. **The question paper is to be returned with the Answer Script.**

Key Examination Protocol

1. Students please note that academic dishonesty (or cheating) includes but is not limited to plagiarism, collusion, falsification, replication, taking unauthorised notes or devices into an examination, obtaining an unauthorised copy of the examination paper, communicating or trying to communicate with another candidate during the examination, and being a party to impersonation in relation to an examination.
2. The above mentioned and any other actions which compromise the integrity of the academic evaluation process will be fully investigated and addressed in accordance with UTT's academic regulations.
3. Please be reminded that speaking without the Invigilator's permission is **NOT** allowed.

Section A:

1. Given the following numbers below:

46	28	24	6	22	18	10

   a. Show the step by step configuration of the list as an Insertion Sort algorithm is applied. [5]

   b. Write a Java method that accepts an array of numbers and performs the Insertion Sort algorithm on the list. [5]

2. Write a Java recursive ***BinarySearch*** method that returns the position of a search key argument in the array argument. The method should accept the following parameters
   a. int binarySearch (int list[], int lo, int hi, int searchKey); [5]

3. Write a Java recursive method that accepts a function parameter, say N, and returns the factorial of N. The function header should resemble the following [5]
   a. int factorial (int N)

4. Write a Java class that implements an Abstract Data Type called Queue using static storage mechanism. The class should define the following methods: [20]

   - init(); // Default size 10 // Prepares Queue for data
   - init(int size); // Prepares Queue for data of a specified size
   - add(Object obj); // Adds an object to the Queue
   - Object remove (); // Removes and returns an object from the Queue
   - boolean isEmpty(); // Returns true if the Queue is empty, false otherwise
   - boolean isFull(); // Returns true if the Queue is full, false otherwise
   - Object peek(); // Returns an object from the Queue but does not remove it

Section B:

5. State briefly what is a balanced binary tree and explain briefly how it differs from a Binary Search Tree. [3]

6. Given the following values below:

67	75	33	52	22	27	5	105	72	64	40	37	24	46

   a. Construct a Binary Search Tree. Using **67** as the root. [4]

   b. Display the in-order traversal [2]

   c. Display the pre-order traversal [2]

   d. Display the post-order traversal [2]

   e. Show the tree after 40 is deleted [2]

7. Write a recursive Java method to remove a value in the Binary Search Tree. The following function header may be used: [5]

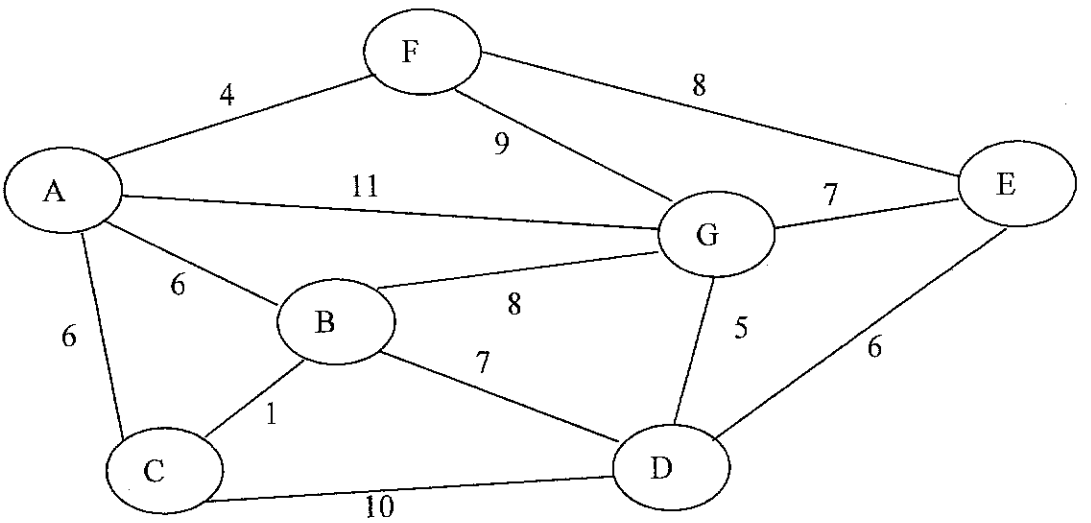
   a. void removeBST (Node root, int data)

8. Write an iterative Java method to traverse a Binary Search Tree and display the data in in-order form. [5]

Section C:

9. Clearly compare and contrast the following terms: [5]
   a. Binary Tree **vs.** Graph
   b. Adjacency List **vs.** Adjacency Matrix
   c. Dijkstra's algorithm **vs.** Kruskal's algorithm.

10. Using the following graph:



   a. Draw the breadth-first and depth-first traversals of the graph starting at node A. Edges are processed in non decreasing order. Clearly show your steps. [10]

   b. Draw the minimum cost spanning tree using Prim's algorithm. At each stage show the connected sub-graphs represented as a collection of disjoint sets. [10]

END OF PAPER