UNIVERSITY *of* LIMERICK

OLLSCOIL LUIMNIGH

FACULTY *of* SCIENCE *and* ENGINEERING

Department of Computer Science
and Information Systems

Final Assessment Paper

Academic Year:	2013/2014	Semester:	Spring
Module Title:	Data Structures and		
Algorithms	Module Code:	CS4115	
Duration of Exam:	2 hours	Percent of Semester Marks:	60
Lecturer:	P. Healy	Paper marked out of:	60

Instructions to Candidates:

- There are two sections to the paper: Short Questions and Long Questions
- The mark distribution is 15 marks for Short Questions and 45 marks for the Long Questions
- Answer all questions in all sections

Section 1. Short Questions (5 × 3 marks).

- Please put your answers to these questions in the answer book provided to you, labelling your answers 1.1, 1.2, etc.

1. Exponentiation, that is, computing x^n, can be done in $o(n)$ time, true or false?
2. What is $S = \sum_{i=2}^{\infty} \frac{1}{2^i}$?
3. In an AVL tree we have seen that a left-left imbalance can be fixed by a single call to the `rotate()` member function of the AVL class; in your answer books give an example of this case *and* how the `rotate()` function should be called in this case; then show how two calls to `rotate()` would fix a left-right imbalance (the double rotation case).
4. Of *all* of the sorting algorithms you know about what would be the most appropriate if you were told that there was only one element out of place in a sequence of integers? What will be the running time of the algorithm in this case? An element is out of place if deleting it from the sequence results in the new sequence being entirely in order.
5. If a graph is extremely dense what is the running time of Depth First Search?

Section 2. Long Questions (45 marks).

- Please put your answers to these questions in the answer book provided to you
- Label your answers 2.1, 2.2, and 2.3 in your answer books

1. Fibonacci number generation. **(15 marks.)**

   (a) Give an $o(n)$-time algorithm for calculating A^n where A is a 2×2 matrix. What is the running time of your algorithm? (5 marks.)

   (b) The Fibonacci number sequence is given by

   $$F_n = F_{n-1} + F_{n-2}, \quad n \geq 2, \quad F_0 = F_1 = 1$$

   Suppose we write two consecutive Fibonacci numbers in the form of a 2×1 matrix, f where

   $$f = \begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix}$$

   show that there is a 2×2 matrix, A, so that when f is multiplied by A we get

   $$A \begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} F_{n+1} \\ F_n \end{pmatrix}$$

   Hint: What is F_{n+1} in terms of F_n and F_{n-1}? (5 marks.)

   (c) Use the previous two parts to show that F_n can be computed in $o(n)$-time. What is the running time of your algorithm? (5 marks.)

2. Binary Trees and Binary Search Trees **(15 marks.)**

   (a) We have seen that a Binary Search Tree can degenerate into a linked list if it is built by successively inserting the elements smallest to largest (or vice versa).

   Give another example of a sequence of insertions where the height of the resulting tree is n. (6 marks.)

   (b) In addition to the pre-order and post-order traversals that exist for a tree of arbitrary degree a binary tree has a third, in-order, possible traversal. Are any of these traversals sufficient on their own to accurately reconstruct a binary tree from its traversal? What about the output from two traversals? Are all three necessary, or is it even possible?

   In the "no" cases give an example of two trees with the same traversal(s); in the "yes" case (if it exists) give a carefully argued algorithm for how to reconstruct the tree. (9 marks.)

3. Graph Algorithms. **(15 marks.)**

In the labs we have seen the close relationship that exists between a graph and a matrix. If an undirected (and unweighted) graph, G, has adjacency matrix, A, then $A[i, j] = 1$ if and only if vertex i is connected to vertex j by an edge. $A^2[i, j]$ denotes the i, j entry of A^2.

(a) Draw a graph G and show its adjacency matrix, A. (2 marks.)

(b) Show that $A^2[i, j] \geq 1$ if and only if there is some vertex k so that the edges (i, k) and (k, j) exist in G. (3 marks.)

(c) In the previous part could $A^2[i, j] > 1$ hold? (2 marks.)

(d) What is the significance of $A^2[i, i]$? (2 marks.)

(e) All of the above relies on G being undirected. If edges are directed then A will no longer be symmetric. What, now, is the meaning of $A^k[i, i]$ in the case of G being directed? (3 marks.)

(f) Hence, give an algorithm to detect if a graph is strongly connected. You can be as inefficient as you like. Recall a graph is strongly connected if every pair of vertices has a directed path between them ("is reachable"). (3 marks.)