UNIVERSITY *of* LIMERICK

OLLSCOIL LUIMNIGH

FACULTY *of* SCIENCE *and* ENGINEERING

Department of Computer Science
and Information Systems

Final Assessment Paper

Academic Year:	2014/2015	Semester:	Autumn
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 with an AVL tree that has three nodes corresponding to the numbers 1, 2 and 3 *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 in a graph with n nodes and m edges?

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. Sorting Algorithms. **(15 marks.)**

   (a) Use radix sort to sort the words of the phrase:

   ```
   Nae!  The cat is on the ice hat.
   ```

   You should ignore punctuation characters (like '!' and '.') but your sort should take account of case; lower case letters should sort earlier than their uppercase equivalent uppercase. So given the letters 'b', 'A' and 'a', the correct sorted order would be 'a', 'A' and 'b'.

   Show the sort status after each pass.

   How many buckets and passes are required? What is the running time of your algorithm in this case and in the general case of radix sort? (10 marks.)

   (b) Although it is not used generally radix sort has certain niche applications. One such situation arises when ordering the vertices of a graph. Suppose, in a computer dating agency, a client is presented with a set of possible matches (say at most 50) and they give each a preference 1 - 50. The computer then needs to sort these possible matches in order of the stated preferences.

   Why would this be a good situation to use radix sort? (5 marks.)

2. Binary- and d-Heaps **(15 marks.)**

   (a) How many nodes are in the heap shown in Figure 1? Justify your answer. (5 marks.)

   (b) As chief algorithmist at a big data company you are asked to develop a new Abstract Data Type called a Querity Prue (QP). The operations it should support are `insert()` and `findMin()`. What asymptotic running times can you give for this ADT? What will be your implementation? (4 marks.)

   (c) We have seen that a binary heap is a data structure that can support the `insert()` and `deleteMin()` operations required of the Priority Queue ADT in $\mathcal{O}(\log n)$-time. What are the comparable running times for a d-heap? (3 marks.)

   (d) In d-heaps, the second last child of node i is at position di. Use this fact to derive an expression for the position of the parent of node i. (3 marks.)

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

   A team championship is decided each year by putting all of the teams' names into a hat – there are n of them. The first two names drawn out of the hat play each other; so do the next two names drawn, and so on until there are no names left in the hat. These matches make up Round 1 of the championship. The losers of these matches are eliminated, the winners go into the hat again and the process repeats, round after round until there is just one overall champion.

   In all of the following questions please give *exact* answers, or as exact as the information given allows you to conclude. No Big-Oh, please.

   (a) If n is an exact power of 2 (5 marks.)

   - over the entire championship, how many games will be played?
   - how many rounds will be needed to determine the overall winner?
   - how many matches will the champion play?

   (b) It would be more usual for the number of teams in the tournament *not* to be an exact power of 2 and might even be an odd number. Answer the three questions of the previous part for general values of n. (5 marks.)

   (c) In some competitions the "stronger" teams may not be required to play in early rounds. Suppose that the n teams are separated into n_1 "weak" teams, who have to play from the beginning and n_2 "strong" teams who are allowed enter the competition at Round 3. Answer the same three questions in this case. Make no assumptions about the oddness/evenness of n_1, n_2. (5 marks.)

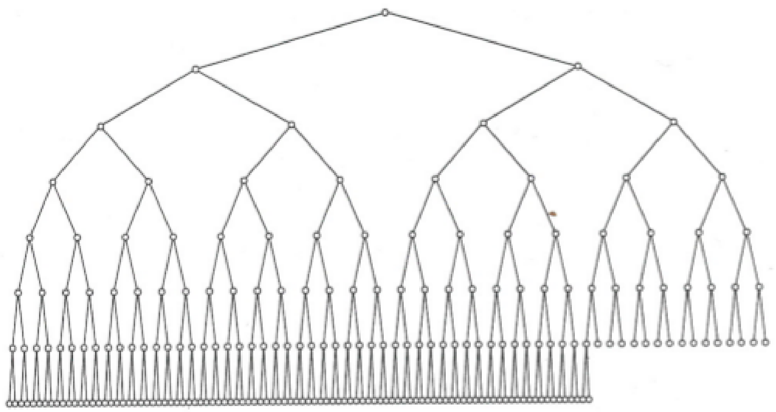

Figure 1: A binary heap.