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:	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.
- Justify all answers; a justified answer is better than a guessed answer.

1. If $f(n) = 2n - 3$ then $f(n) = O(n)$ *and* $f(n) = O(n^2)$. True or false?
2. How many passes of radix sort are required to sort an array of integers if we are limited to using 256 buckets on any pass? Justify your answer.
3. Binary search is both $O(n)$ and $o(n)$. True or false? A justified answer is better than a guessed answer.
4. A connected graph has at least m edges. What value is m?
5. Of *all* of the sorting algorithms you know about which 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.

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. Exponential Growth Rates. **(15 marks.)**

   (a) The Fibonacci numbers are given by the formula

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

   Use induction to show that $F_n \leq 2^{n-2}$ for almost all n. Pay particular attention in your argument to the base case. (5 marks.)

   (b) What is the largest number of nodes that a binary tree of height h (having levels $0, \ldots, h$) can have? (5 marks.)

   (c) Carefully justify your answer of the previous part. Your answer should argue that the tree cannot possibly accommodate any more nodes. (5 marks.)

2. Code and Algorithm Analysis **(15 marks.)**

   (a) The binomial coefficient, $C(n,k) = \binom{n}{k}$ which counts the "number of ways to choose k items from a set of n items" has many uses. It is defined as

   $$\begin{aligned}\binom{n}{k} &= \frac{n!}{k!(n-k)!} \\ &= \frac{n \times (n-1) \times (n-2) \times \cdots (n-k+1)}{1 \times 2 \times \cdots \times k}\end{aligned}$$

   One application of the binomial formula is in computing Bézier curves for graphics. The code below, for a given value n, computes all binomial coefficients $\binom{n}{k}, 0 \leq k \leq n$ and stores them in the array C.

```
/*  Compute binomial coefficients C for given value of n.  */
void binomialCoeffs(int n, int * C)
{
   int k, j;

   for (k = 0;  k <= n;  k++) {
      /*  Compute n!/(k!(n - k)!).  */
      C [k] = 1;
      for (j = n;  j >= k + 1;  j--)
        C [k] *= j;
      for (j = k;  j >= 1;  j--)
        C [k] /= j;
   }
}
```

   The code is taken from a commonly used graphics textbook and, while good enough for teaching purposes, the code has a flaw that prevents it from being generally usable. The flaw arises due to integer overflow in computing the numerator.

i. Explain how integer overflow prevents the correct calculation of some binomial values, even if the answer can be represented safely in the hardware. Will moving to new hardware with a longer word size make the problem go away? (3 marks.)

ii. Rewrite the function so that the problem is avoided (3 marks.)

iii. Apart from the issue of integer overflow the function is inefficient in that it repeats computations. Rewrite the function so that it is as efficient as you can make it. (3 marks.)

(b) While it is true that an algorithm that performs $n^2/2$ operations on n inputs is going to run faster than an algorithm that performs n^2 operations, give **two** reasons, with justification, why we really shouldn't get too excited about the former over the latter. (6 marks.)

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

(a) Discuss how you could use an algorithm we have seen in labs to tell what nodes in a graph are exactly 4 edges away from a given node. (5 marks.)

(b) The Erdös Numbering Problem is to determine for a given computer scientist how *closely related* the researcher is to Paul Erdös, often considered to be one of the greatest computer scientists ever.

The Erdös Number (EN) of Erdös himself is 0; the Erdös number of an author is defined to be 1 if the author has co-written a paper in a scholarly journal with Erdös; otherwise, to determine an author's EN, find the lowest EN amongst all the people he has ever co-written a paper with and add 1 to that.

So, if author "Joe Bloggs" has EN 7, then "Seoirse de Blogg", a colleague and co-author of Joe Bloggs', will have EN *at most* 8. Note that his EN could possibly be lower than 8 if he has worked with somebody else who is more closely related to Erdös.

Using data structures and algorithms that we have studied in class develop an algorithm that reads a database of co-authors and determines the EN of some specified author. Your answer should be clear instructions along the lines of pseudo-code that could be turned into a program by a third person.

What is the worst-case running time of your algorithm? What are its space requirements? (10 marks.)