UNIVERSITY *of* LIMERICK

OLLSCOIL LUIMNIGH

COLLEGE *of* INFORMATICS *and* ELECTRONICS

Department of Computer Science
and Information Systems

Final Assessment Paper

Academic Year:	1999/2000	Semester:	Autumn
Module Title:	Data Structures and Algorithms	Module Code:	CS4115
Duration of Exam:	$2\frac{1}{2}$ hours	Percent of Semester Marks:	60
Lecturer:	P. Healy	Paper marked out of:	100

Instructions to Candidates:

- There are three sections to the paper: Multiple Choice Questions, Short Questions and Long Questions
- The mark distribution is 55 marks for Multiple Choice Questions, 15 marks for Short Questions and 30 marks for the Long Questions
- Answer all questions in all sections

Section 1. Multiple Choice Answers (55 marks in total).

Use the machine-readable multiple-choice question grid that has been provided to answer these questions. Please completely mark in black exactly one circle on the grid for each answer.

1. On the first day of Christmas,
my true love sent to me
A partridge in a pear tree.
On the second day of Christmas,
my true love sent to me
Two Zetor tractors, and
A partridge in a pear tree.
On the third day of Christmas ...

How many lines would be in such a "poem" if it ran for 365 days instead of the usual 12?

(a) $\frac{365\times366}{2} + 2 * 365$
(b) $\frac{367\times368}{2} - 3$
(c) Neither of the above
(d) Both of the above

2. Which of the answers below is $\sum_{i=2}^{n} 4^i$?

(a) $\frac{1}{3}(4^{n-1} - 1)$
(b) $\frac{1}{3}(4^n - 1)$
(c) $\frac{16}{3}(4^{n-1} - 1)$
(d) None of the above

3. Which of the answers below best approximates $\sum_{i=0}^{n} i^5$?

(a) $O(i^5)$
(b) $O(n^4)$
(c) $O(n^6)$
(d) $O(n^5)$

4. An $O(n)$-time algorithm sorting integers in the range $[0, 2^{32} - 1]$ is possible by making 4 passes of *radix sort*, with buckets of size

(a) $2^{32}/4$
(b) $2^{32}/2^4$
(c) 2^8
(d) 2^4

(please turn over)

5. The time-complexity of exponentiation to the power of n is (most precisely)

(a) $O(n)$
(b) $O(\log n)$
(c) $o(n)$
(d) $\Theta(n)$

6. What is the desired upper limit on the load factor, λ, for *open* hashing?

(a) $\lambda \leq 2.0$
(b) $\lambda \leq 1.0$
(c) $\lambda \leq 0.5$
(d) $\lambda \approx 0.0$

7. How many bits are required per node to store the height of a node in an n-node AVL tree?

(a) n
(b) $\log n$
(c) $\log \log n$
(d) $\log^2 n$

8. How many nodes are on the bottom layer, h, of a *complete* binary tree?

(a) at least 2^h
(b) at most 2^h
(c) exactly 2^h
(d) none of the above

9. An *inversion* in a list of numbers is an occurrence of a pair of numbers not in order. What is the *expected* number of inversions in a list of n numbers?

(a) $\frac{n(n-1)}{4}$
(b) $\frac{n(n-1)}{2}$
(c) $\frac{n(n+1)}{4}$
(d) $\frac{n(n+1)}{2}$

10. Suppose we exchange elements $A[i]$ and $A[i+k]$ of an array, which were originally out of order. How many inversions does this remove from the array, *at most*?

(a) 1
(b) 2
(c) $2k - 1$
(d) $2k + 1$

11. Which of the following sorting algorithms is the odd one out?

(a) Insertion sort
(b) Selection sort
(c) Bubble sort
(d) Mergesort

12. Although Mergesort has a better worst-case running time guarantee than Quicksort, it is not the sorting algorithm of choice. This is because

(a) It can be outperformed for small values by, say, insertion sort
(b) It has high overheads (costs) associated with it
(c) It needs to have all of its input in primary memory
(d) Its performance degrades significantly when sorting non-integers

13. Embedded in Shellsort is what sorting algorithm?

(a) Insertion sort
(b) Bucket sort
(c) Heapsort
(d) Bubblesort

14. For Shellsort, a *bad* sequence of increments, h_k, is one where

(a) Each of the increments is a multiple of the previous
(b) Each of the increments is a prime number
(c) The increments pairwise have no common factors
(d) No set of positions of the array will be in the same subarray too often

15. Finding the median ($\frac{n}{2}$th element) of an n element set can be done in average time

(a) $O(1)$
(b) $O(n)$
(c) $O(n \log n)$
(d) $O(n^2)$

16. An adjacency array would be an appropriate representation of an n-vertex graph if the number of edges was

(a) $\Omega(n^2)$
(b) $O(n)$
(c) $o(n)$
(d) $O(1)$

17. The running time of Depth First Search on a graph, G, is

(a) $O(|V| \times |E|)$
(b) $O(\min(|V|, |E|))$
(c) $O(|V| + |E|)$
(d) $O(\max(|V|, |E|))$

18. A graph, G, can be tested for *acyclicity* in time

(a) $O(|V| \times |E|)$
(b) $O(\min(|V|, |E|))$
(c) $O(|V| + |E|)$
(d) $O(\max(|V|, |E|))$

19. "Every vertex has even degree" is a _____ condition for an Eulerian Circuit. What goes in the blank?

(a) necessary
(b) sufficient
(c) both necessary and sufficient
(d) neither necessary nor sufficient

20. The Depth First Search-based algorithm for *biconnectivity* is an example of both _____ and _____ processing of nodes. The blanks are

(a) inorder, preorder
(b) inorder, postorder
(c) preorder, postorder
(d) inorder, postalorder

Section 2. Fill in the blank (5 × 3 marks).

- Please put your answers to these questions in the answer book provided to you, labelling your answers 2.1, 2.2, etc.

1. In open hashing, each linked list should have _____ element(s) on *average*.

2. The maximum number of nodes in a binary tree of height h is _____.

3. A large number of deletions in an open hash table can cause the hash table can cause the table ot be fairly empty, which wastes space. In this case, we can rehash to a table half as large large. Assuming that we rehash to a larger table when there are twice as many elements as the table size, how empty should an open table be before we rehash to a smaller one? _____

4. The function `percolateDown()` takes time _____ in the worst case; using $O(n)$ calls to `percolateDown()` the `buildHeap()` function takes _____ worst-case time.

5. In the Depth First Search tree T of a graph G, a *back edge* indicates the presence of a _____.

Section 3. Long Questions (30 marks).

- Please put your answers to these questions in the answer book provided to you
- Label your answers 3.1, 3.2 and 3.3 in your answer books, please

1. Give a high-level (but precise) algorithm to multiply two polynomials, using a linked list implementation. Your algorithm should run in $O(m^2 n)$ time, where m is the size of the smaller polynomial. Make sure that the output is sorted by exponent and that there is at most one term for any power. (15 marks.)
Hint 1: Drawing pictures of how your algorithm works will more than likely help me understand your algorithm. This is a good thing.
Hint 2: You should not use any explicit sorting function.

2. Show the result of inserting 3, 1, 4, 6 (in that order) into an empty binary tree. Now show the result of inserting 9, 2, 5, 7 (in that order) into this tree. (5 marks.)

3. Prove that every graph must have an even number of vertices of odd-degree. (10 marks.)