UNIVERSITY of LIMERICK

OLLSCOIL LUIMNIGH

COLLEGE of INFORMATICS and ELECTRONICS

Department of Computer Science
and Information Systems

Annual Repeats Assessment Paper

Academic Year:	2004/2005	Semester:	Autumn
Module Title:	Data Structures and Algorithms	Module Code:	CS4115
Duration of Exam:	$2\frac{1}{2}$ hours	Percent of Semester Marks:	65/100
Lecturer:	P. Healy	Paper marked out of:	100

Instructions to Candidates:

- There are two sections to the paper: Short Questions and Long Questions
- The mark distribution is 40 marks for Short Questions and 60 marks for the Long Questions
- Paper counts for 65% for students with an I-grade; paper counts for 100% for those capped at C3
- Answer all questions in all sections
- **You *must* return this paper with your answer book and bubble sheet**

Section 1. Short Questions (10 × 4 marks).

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

1. The height of an AVL tree is no worse than ________ times the optimal height.
2. Give the recurrence relation for N_h, the number of nodes in the worst possible AVL tree of height h ____________________________.
3. In a d-heap (a heap where each node can have at most d children), what are the locations of a node's children? The root node of the heap is at location 1. ______________________
4. Sorting is possible in $o(n \log n)$-time with _____ sort.
5. If a graph has negative edge costs then the "Big-Oh" running time of the shortest path algorithm increases to _________.
6. In open hashing, each linked list should have ___ element(s) on *average*.
7. The maximum number of nodes in a binary tree of height h is ________.
8. A large number of deletions in an open hash table can cause the hash table can cause the table to be fairly empty, which wastes space. In this case, we can rehash to a table half as 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? ____
9. When implementing the heap data structure, the function `percolateDown()` takes time ________ in the worst case; using $O(n)$ calls to `percolateDown()` the `buildHeap()` function takes _____ worst-case time.
10. In the Depth First Search tree T of a graph G, a *back edge* indicates the presence of a _____.

Section 2. Long Questions (60 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. **(20 marks.)**

   (i) Show the result of inserting $3, 1, 4, 6, 9, 2, 5, 7$ into an initially empty binary search tree.

   (ii) Show the result of deleting the root.

2. **(20 marks.)**

   Given the input { 4371, 1323, 6173, 4199, 4344, 9679, 1989 } and a hash function $h(x) = x \bmod 10$

   With $hsize = 11$, show the resulting tables under

   (a) open hashing

   (b) closed hashing using linear probing

   (c) closed hashing using quadratic probing

   (d) closed hashing with secondary hash function $h_2(x) = 7 - (x \bmod 7)$

   (e) following rehashing what will be the table look like after closed hashing using linear probing

   For each of the inputs values, x, the list of values $x \bmod 7$ is, respectively { 3, 0, 6, 6, 4, 5, 1 }.

3. **(20 marks.)**

   A *bipartite* graph is a graph, $G = (V, E)$, such that V can be partitioned in to two disjoint sets V_1 and V_2 and no edge has its two endpoints in the same set. A picture of a bipartite graph is shown below in Figure 1.

   Prove that a graph is bipartite if and only if it contains no cycles with an odd number of edges.

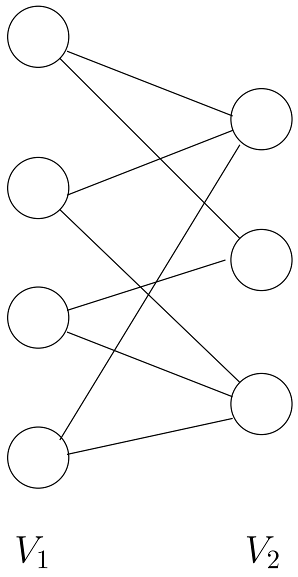

Figure 1: A bipartite graph.