THE UNIVERSITY OF THE WEST INDIES
ST. AUGUSTINE

EXAMINATIONS OF

Code and Name of Course: **COMP 2000 – Data Structures** Paper:

Date and Time: April/May 2010 Duration: **2 hours**

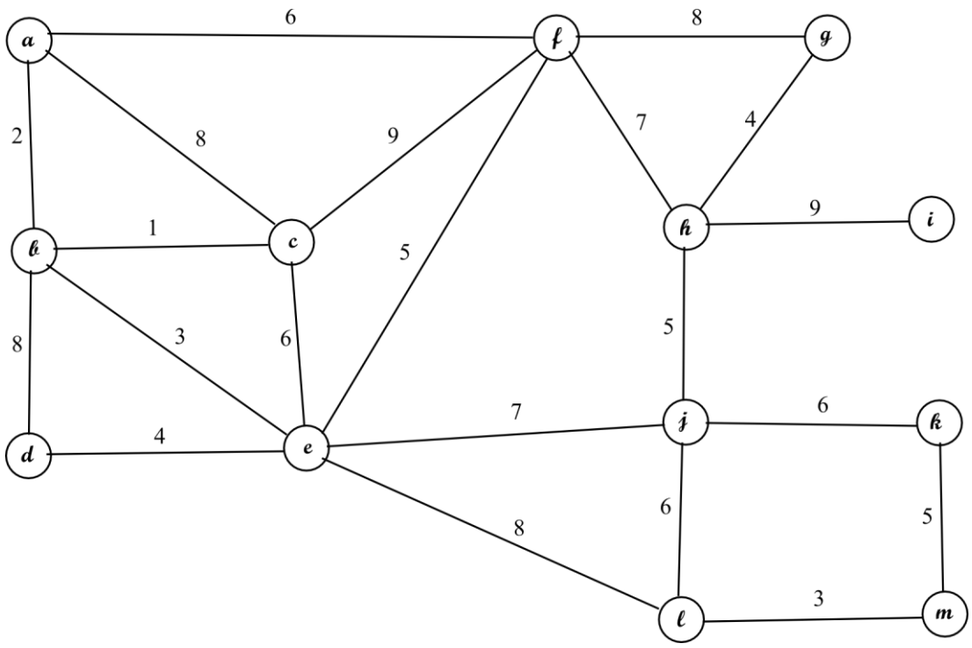
INSTRUCTIONS TO CANDIDATES: This paper has **4** pages and **6** questions

Answer ALL questions
Questions are not equally weighted

PLEASE TURN TO THE NEXT PAGE

Question 1: Graphs and Matrices

For the following graph:



(a)

(i) Show how the graph will be represented using an adjacency list representation. **[2]**

(ii) If a node label may be stored as an ASCII character, an integer is four bytes in size and a pointer is four bytes in size, what would be the size in bytes of the adjacency list representation? **[3]**

(b)

(i) Draw a single source shortest path tree with node ***a*** as source. (For ease of drawing, use the same arrangement of nodes as above, but include only path edges.) **[5]**

(ii) Indicate the order in which edges are included in the tree using Dijkstra's algorithm. **[5]**

Question 1 (cont'd)

(c) Since the graph is an undirected graph, its adjacency matrix will be symmetric, and lower-packed storage can be used to reduce the size of the stored matrix.

(i) Given that an integer is four bytes in size, what would be the size in bytes of the adjacency matrix if lower-packed storage is used? **[3]**

(ii) Write a function in pseudocode that takes the lower-packed array and two edge labels as arguments and returns the weight of the edge joining them (where a weight of zero indicates that there is no edge joining the nodes). **[5]**

(d) (i) Show how the adjacency matrix for the graph could be stored as a sparse matrix using compressed row storage. **[5]**

(ii) How much memory will be used to store the sparse array using compressed row storage? **[2]**

[Total 30]

Question 2: Trees

(a) Without assuming any particular storage method for a tree, write algorithms in pseudocode for doing:

(i) a pre-order traversal of a tree; **[1]**

(ii) an in-order traversal of a tree; **[1]**

(iii) a post-order traversal of a tree. **[1]**

(b) Without assuming any particular storage method for a **binary search tree**, write in pseudocode, algorithms for the following, such that the tree remains a binary search tree:

(i) finding a node; **[3]**

(ii) inserting a node; **[3]**

(iii) deleting a node. **[6]**

[Total 15]

Question 3: Hash Tables

Show the resulting contents of a hash table of 10 after the elements

6, 24, 28, 64, 36

are inserted in the order given, using the primary hash function, $h_1(k) = k \% 10$, and:

(a) linear probing. **[2]**

(b) double hashing with the secondary hash function, $h_2(k) = (1+k \% 3)$. **[3]**

[Total 5]

Question 4: Heaps

With reference to the following array of integers

6	11	28	65	72	14	23	64	97	36

(a) Draw the nearly complete binary tree represented by this array. **[2]**

(b)

(i) Write a function in pseudocode that rearranges the elements of the nearly complete binary tree into a max-heap. **[2]**

(ii) Show by a series of drawings of the array contents how the array is rearranged by the algorithm in part (b)(i). **[3]**

(c) Explain with the aid of a series of drawings of the array contents how the function in part (b)(i) can be used to perform a heapsort on the array elements. **[3]**

[Total 10]

Question 5: Sorting and Searching

(a) Write an algorithm in pseudocode for the following sorting techniques:

(i) quicksort; **[2]**

(ii) mergesort; **[2]**

(iii) shellsort. **[2]**

(b) Using the algorithms specified in part (a) of this question, show by means of a series of drawings of the array contents, how the array

6	11	28	65	72	14	23	64	97	36

will change as it goes through a

(i) quicksort; **[3]**

(ii) mergesort; **[3]**

(iii) shellsort using increments of 1, 3 and 5. **[3]**

[Total 15]

End of Question Paper