SPEC
(39)

THE UNIVERSITY OF THE WEST INDIES
ST. AUGUSTINE

EXAMINATIONS OF APRIL/MAY 2012

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

Date and Time: Friday 11th May 2012 9 am. Duration: **2 hours**

INSTRUCTIONS TO CANDIDATES: This paper has **4** pages and **5** questions

Answer ALL questions

Questions are not equally weighted

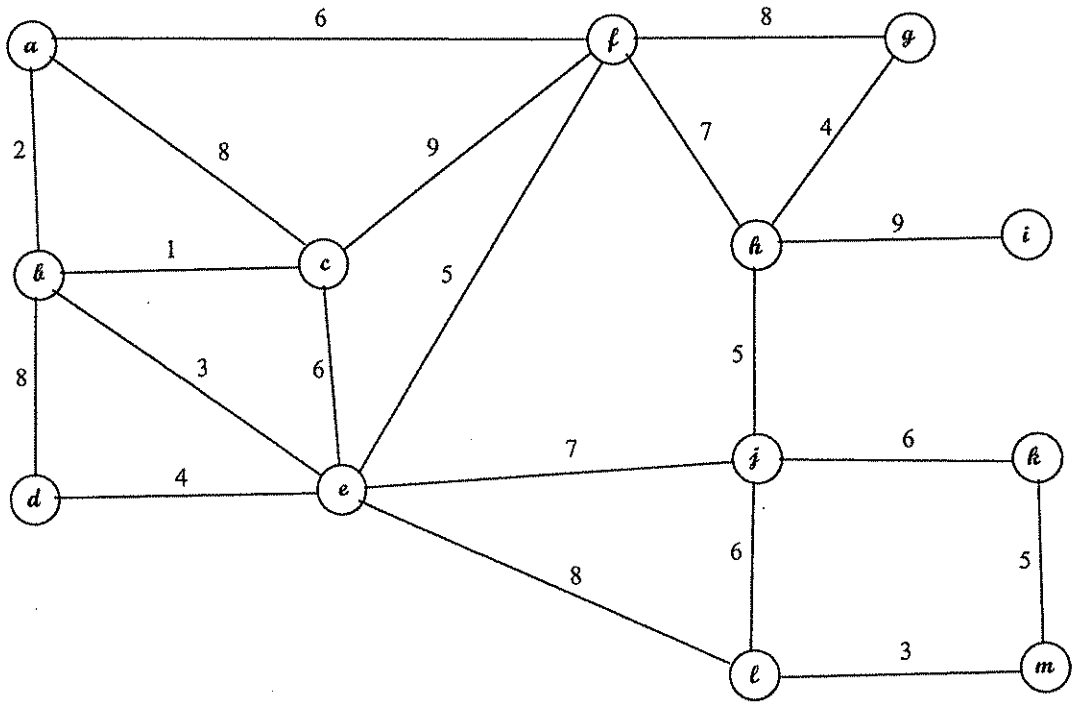
Please follow the instructions on the answer booklet. Specifically,

- **There is a place at the top of each page for the number of the question that is answered on that page.**
- **Pages must also be numbered so that attached pages can be placed in the correct order.**
- **If parts of questions are done out of order, then they must be properly numbered and the order in which the parts were done must be entered in the space provided for this purpose at the bottom of the cover of the answer booklet.**

QUESTIONS BEGIN ON THE NEXT PAGE

Question 1: Graphs, Trees and Matrices

For the following graph:



(a)

For each of the following options, state two considerations that would move you to implement a graph using the chosen data structure:

(i) an adajency list rather than as an adjacency matrix; **[2]**

(ii) an adjacency matrix rather than as an adjacency list. **[2]**

(b) Assume that a node label may be stored as an ASCII character string, an integer is four bytes in size and a pointer is four bytes in size. Further assume that you have chosen to store the graph using an adjacency matrix representation and use a binary search tree (BST) to store each node label and the matrix index associated with that label.

(i) Show data declarations in C that you would use for implementing this BST using a linked structure? **[3]**

(ii) Show in pseudocode that algorithm that you would use to search this BST for the matrix index associated with a given node label. **[3]**

Question 1: (cont'd)

(iii) 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. Show in pseudocode the algorithm you would use to access elements in the adjacency matrix if lower-packed storage is used. Assume that you already know the row and column indices of the elements to be accessed and the algorithm has to access the required element in the lower packed storage structure. **[3]**

(iv) What would be the size in bytes of the lower-packed matrix if only the edge weights are stored as elements of the matrix? **[2]**

[Total 15]

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; **[2]**

(ii) inserting a node; **[2]**

(iii) deleting a node. **[3]**

[Total 10]

Question 3: Hash Tables

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

6, 36, 28, 27, 65

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

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

(b) quadratic probing with a linear probing term coefficient (c_1) of 1 and a quadratic probing term coefficient (c_2) of 2. The primary hash function should be $h'(k) = k$. **[3]**

[Total 5]

Question 4: Heaps, Heapsort

With reference to the following array of integers:

65	97	60	25	23	72	36	32	12	8

(a)

(i) Draw the nearly complete binary tree represented by this array. **[1]**

(ii) Write an iterative algorithm in pseudo-code that rearranges the elements of an array representing a nearly complete binary tree into a min-heap. **[2]**

(ii) Show by a series of drawings of the array contents how the array above is rearranged on each iteration of the algorithm in part (a)(ii) . **[3]**

(c) Explain with the aid of a series of drawings of the array contents how the above array is rearranged on each iteration of a heapsort. The sequence of values should be sorted in descending order. **[3]**

[Total 9]

Question 5: Sorting

In this question, assume all sorting is done in ascending order.

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

(i) quicksort; **[4]**

(ii) mergesort; **[4]**

(iii) shellsort. **[4]**

(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

65	97	60	25	23	72	36	32	12	8

will change as it goes through a

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

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

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

[Total 21]

End of Question Paper