UNIVERSITY OF SWAZILAND

SUPPLEMENTARY EXAMINATION 2006

Title of paper: DATA STRUCTURES

Course number: CS342

Time allowed: Three (3) hours

Instructions: Answer any five (5) of the six (6) questions.

This examination paper should not be opened until permission has been granted by the invigilator.

Question 1

a) State precisely the meaning of the statement: *f(n) is O(g(n))*

[4]

b) Justify the following statement without directly invoking the definition of big-O notation. State the applicable rules of justification.

$N^2 + log^3 N$ is $O(2^N + N^3)$

[8]

c) Compute the approximate number of operations required to retrieve the N^{th} element of an array and of a linked list. What can be concluded from these approximations regarding the relative speed of element-retrieval in arrays and linked lists?

[4]

d) Compute the approximate amount of memory required to store N integers in an array and in a linked list. What can be concluded from these approximations regarding the relative memory usage in arrays and linked lists?

[4]

Question 2

a) Describe the main operations of the queue ADT.

[6]

b) Write an algorithm that will take two inputs — a queue and its length — and insert next to each of the queue's elements, a duplicate of that element. For example, if the queue had length 3 and contained elements [1, 10, -3] (with 1 at the front), then the algorithm will change the queue to [1, 1, 10, 10, -3, -3]. Your algorithm may not make use of any data structures other than the given queue. Briefly explain how your algorithm works.

[6]

c) Give an array-based (not list-based) implementation of the stack ADT.

[8]

Question 3

a) Generally, in what situations is it preferable to use stacks rather than lists. Furthermore, in what situations are lists preferable?

[4]

b) Consider the problem of determining whether or not a given list contains any duplicates (more than one copy of any list member).

i. Write down an algorithm that inputs a list and returns a Boolean value stating whether or not any two members of the list are identical.

[10]

ii. How may the size of this problem be quantified?

[1]

iii. Compute the approximate number of operations performed by the algorithm, with respect to problem size, and hence its big-O time complexity.

[5]

Question 4

a) Give an implementation of the queue ADT that makes use of the list ADT.

[8]

b) Write an algorithm that, given an integer n and a pointer to the initial node of a linked list, will return the value found in the n-th node. E.g. if n=2, the value found in the second node will be returned. Assume that n is positive and not greater than the length of the list.

[5]

c) Write an algorithm that, given pointers to the initial nodes of two linked lists, will append the nodes of the second list to the end of the first list. Assume that neither linked list is empty. E.g. if the first list consisted of [1, 2, 3] and the second consisted of [5, 6], then the first list would be changed to [1, 2, 3, 5, 6].

[7]

Question 5

a) Define the following terms in relation to trees:

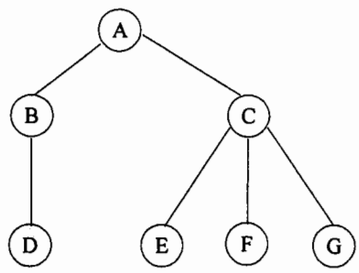
[3]

i. Root node
ii. Child node
iii. Sibling node

b) Give the algorithm for traversing a general tree in preorder.

[5]

c) Consider the tree below:



i. What is the depth of the tree?

[1]

ii. Give the sequence of nodes visited by preorder, inorder and postorder traversals.

[6]

iii. Write an algorithm that will construct the tree.

[5]

Question 6

a) With the aid of a graph diagram, distinguish between depth-first and breadth-first traversal.

[5]

b) Write a non-recursive algorithm that, given a graph vertex, will traverse the graph in depth-first order commencing at the given vertex.

[10]

c) Explain how, by modifying slightly the depth-first traversal algorithm given above (part b), breadth-first traversal may be achieved.

[5]