UNIVERSITY OF SWAZILAND

FINAL 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.

$(N^2 + N + 1)\cdot(log^2 N)$ is $O(N^3)$

[8]

c)

i. Using Pascal or other suitable notation, define a data type for linked-list nodes.

[2]

ii. Based on your answer to the above (part c.i), draw a memory diagram of a linked list consisting of 3 values. Your diagram must show a series of labelled memory locations and their contents.

[4]

d) Explain the main advantage of linked lists over arrays.

[2]

Question 2

a) Generally, in what situations is it preferable to use stacks or queues rather than lists?

[2]

b) Write an algorithm that, given a queue and a stack, will reverse the order of the queue's elements with the aid of the stack. Briefly explain how your algorithm works.

[6]

c) Explain the ways in which pointer-based implementation of stacks are more efficient than array-based implementations.

[4]

d) Give an implementation of the stack ADT that makes use of the list ADT.

[8]

Question 3

a) Write an algorithm that, given a value and a list, will remove from the list all copies of the given number. Briefly explain how your algorithm works.

[7]

b) Consider the problem of finding the average of a given list of numbers.

i. Write down an algorithm that inputs a list of numbers and returns their average.

[6]

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

[1]

iii. Compute the algorithm's approximate memory consumption with respect to problem size, and hence its big-O space complexity.

[2]

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

[4]

Question 4

a) Fully describe each operation of the list ADT.

[8]

b) Write an algorithm that, given two queues of numbers, each sorted in ascending order (i.e. each member is greater than or equal to the members in front of it), will construct and return a new queue, also sorted in ascending order, consisting of all members of the given queues. E.g. if the first queue is [1, 3, 3] (1 in front) and the second is [-3, -2, 0, 1, 4] (-3 in front), then the returned queue will be [-3, -2, 0, 1, 1, 3, 3, 4] (-3 in front).

[12]

Question 5

a) Define the following terms in relation to trees:

[4]

i. Leaf node
ii. Parent node
iii. Sibling node
iv. Depth

b) Draw a binary tree of consisting of 7 nodes and having a depth of 3.

[3]

c) Give the algorithm for traversing a binary tree in inorder.

[4]

d) Fully explain each operation of the binary tree ADT.

[9]

Question 6

a) Discuss the adjacency-matrix and adjacency-list representations of directed graphs, including their relative time and space complexities.

[10]

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

[10]