The University of Nottingham

SCHOOL OF COMPUTER SCIENCE
AND INFORMATION TECHNOLOGY

A LEVEL 2 MODULE, AUTUMN SEMESTER 2007-2008

ALGORITHMS AND DATA STRUCTURES

Time allowed TWO hours

Candidates must NOT start writing their answers until told to do so

Answer QUESTION ONE and THREE other questions

Marks available for sections of questions are shown in brackets in the right-hand margin.

No calculators are permitted in this examination.

Dictionaries are not allowed with one exception. Those whose first language is not English may use a dictionary to translate between that language and English provided that neither language is the subject of this examination.

No electronic devices capable of storing and retrieving text, including electronic dictionaries, may be used.

DO NOT turn examination paper over until instructed to do so

Question 1

This multiple choice question is compulsory. Please choose one answer in every part.

(a) Which of the following is a concrete data structure rather than an abstract data type:

i stack

ii queue

iii priority queue

iv linked list

v binary tree

(4 marks)

(b) Which of the following algorithms uses a divide and conquer strategy:

i bubble sort

ii selection sort

iii insertion sort

iv bucket sort

v merge sort

(3 marks)

(c) Which of the following is a minimal spanning tree of the graph below:

```
          10
    A --------- B
    |\         /
    | \ 12    / 12
10  |  \     /
    |   \   /
    C --- D
       5
```

i a set of edges (A, B), (A, C), (C, D)

ii a set of edges (A, B), (B, D), (C, D)

iii a set of edges (A, B), (A, D), (C, D)

iv a set of edges (A, B), (A, C), (A, D)

v a set of edges (A, C), (A, D), (C, D)

(3 marks)

(d) Which of the following is an invariant of the loop in the search method shown below:

```
boolean search(int[] array, int value){
   for (int j = 0; j < array.length; i++) {
      if (array[j] == value) return true;
   }
   return false;
}
```

i `j` $<$ `array.length`

ii for all array indices `k`: if `k` $<$ `j` then `arr`[`k`] is not equal to `value`

iii for all array indices `k`: if `k` $\leq$ `j` then `arr`[`k`] is not equal to `value`

iv `array`[`j`] $==$ `value`

v `value` is not in the array

(3 marks)

(e) What is the worst case time complexity of the method below (where n is `array.length`):

```
boolean search(int[] array, int value){
   for (int j = 0; j < array.length; i++) {
      if (array[j] == value) return true;
   }
   return false;
}
```

i $O(1)$

ii $O(log_2\ n)$

iii $O(n)$

iv $O(n^2)$

v $O(n + \texttt{value})$

(3 marks)

(f) In-order traversal of a binary search tree visits all nodes in the tree in ascending order of keys. Which of the following defines in-order traversal:

i visit the nodes level by level, starting from the leaves

ii root first, then in-order of the left subtree, then in-order of the right subtree

iii in-order of the left subtree, then root, then in-order of the right subtree

iv in-order of the left subtree, then in-order of the right subtree, then the root

v in-order of the right subtree, then root, then in-order of the left subtree

(3 marks)

(g) Suppose you are running an algorithm on a set of files. The running time of your algorithm is $O(n^3)$, where n is the number of files. For 200 files it takes 1 hour to complete. How long would you expect it to run on 400 files?

i 2 hours

ii 4 hours

iii 6 hours

iv 8 hours

v 60^3 minutes (which is 3600 hours)

(3 marks)

(h) An algorithm's running time is described by the following function: $t(n) = 2n + n * log_2 n$. What is the time complexity of this algorithm:

i $O(1)$

ii $O(log_2 n)$

iii $O(n)$

iv $O(n\ log_2 n)$

v $O(n^2)$

(3 marks)

Question 2

(a) Bounded stacks are stacks which have a fixed maximal size and no items can be pushed onto the stack after it is filled to capacity. Give method signatures and pre- and postconditions for the methods of a Bounded Stack ADT. Assume that items to be pushed on the stack are of type Object. Give methods for pushing items on the stack, popping the stack, peeking at the top of the stack, checking whether the stack is empty or full. (10 marks)

(b) Write a Java class which implements your ADT using an array. You may also need to write classes for exception objects which are thrown when method preconditions do not hold. (15 marks)

Question 3

(a) What is a binary search tree? Explain how items in a binary search tree are ordered and how this ordering is used by a search method. (5 marks)

(b) What is the reason for keeping a binary search tree balanced? Describe two kinds of balanced binary search trees. (5 marks)

(c) What is a 2-4 tree? List the conditions on how many data items a node in a 2-4 tree may contain, how many children it may have, and on the order of the children. (5 marks)

(d) Draw the result of inserting items with keys 10,11,12,13,14,15 and 16 in an empty 2-4 tree. (5 marks)

(e) What is the big O complexity of search in a 2-4 tree? Explain why. (5 marks)

Question 4

Imagine that you want to implement a 'to-do' program. Each day, the program reminds the user of the things they have to do in order of importance. Describe the data structures and algorithms you would employ in the program, and the reasons for your choices. Where appropriate, specify ADTs, pseudocode for significant ADT methods employed, and an outline of the entire program. (25 marks)

Question 5

Consider the graph given by the following adjacency lists:

```
A ---> {B, C, D}
B ---> {D, E}
C ---> {}
D ---> {F}
E ---> {}
F ---> {}
```

(a) Trace a breadth-first traversal of the graph starting from A using a queue. At each step, show which nodes are in the queue. (5 marks)

(b) Trace a depth-first traversal of the graph starting from A using a stack. At each step, show which nodes the stack contains. (5 marks)

(c) Trace a topological sort of the graph. At each step, display the state of the output array of nodes and the nodes and edges which remain in the graph. (7 marks)

(d) Add one more edge to the graph, from F to B. Show the resulting adjacency list. (1 marks)

(e) Using topological sort or DFS, demonstrate how to detect a cycle in a directed graph, using the graph from part (d) as an example. (7 marks)

Question 6

(a) Explain how the quicksort algorithm works. Trace it on the following sequence: `5,3,2,5,4`. Assume that the pivot is always the first element in the range to be sorted. (10 marks)

(b) Give an implementation of quicksort for arrays or for lists in Java or in some other standard programming language. (10 marks)

(c) Describe the worst case performance of quicksort and give big O notation for its running time both in the worst case and in the average case. (5 marks)