THIS IS NOT AN OPEN-BOOK EXAMINATION - CANDIDATES MAY NOT CONSULT ANY REFERENCE MATERIAL DURING THE SITTING.

THE UNIVERSITY OF BIRMINGHAM

MSc in Computer Science

06 02473

SEM 520
FUNDAMENTALS OF COMPUTER SCIENCE

Tuesday 25th May 1999 2.00 pm - 5.00 pm

[Answer ALL Questions]

[Use a Separate Answer Book for EACH SECTION]

Turn Over

THIS IS NOT AN OPEN-BOOK EXAMINATION - CANDIDATES MAY NOT CONSULT ANY REFERENCE MATERIAL DURING THE SITTING.

[Answer ALL Questions]

[Use a Separate Answer Book for THIS SECTION]

Section 1:

1. (a) Explain what are meant by the following terms used when describing the architecture of a computer: instruction, instruction set, machine code program. [3%]

(b) Instructions may be categorised in various ways. Suggest a suitable categorisation and give examples of typical instructions in each of the different categories that you are defining. [4%]

(c) What are the main ways in which the order of execution of consecutively stored program instructions might be altered? [3%]

(d) What is an **interrupt signal** and why is such a signal often used for input/output operations? [2%]

(e) How can multiple interrupts be handled if they occur simultaneously? [2%]

(f) What are the main services provided by an Operating System for the convenience of a programmer? [4%]

(g) In a multi-tasking (multi-processing) environment a particular task (process) may be in any one of several states. Use a state diagram to show the main states that a task (process) may be in and also the principal reasons for transitions between these different states. [4%]

(h) Highlight the main differences between the Ethernet and the Token ring network standards [4%]

(i) What is the Church-Turing thesis and what is meant by the property of algorithms known as universality? [4%]

(j) Outline briefly the basic principles of the Divide and Conquer method of devising efficient algorithms. [4%]

THIS IS NOT AN OPEN-BOOK EXAMINATION - CANDIDATES MAY NOT CONSULT ANY REFERENCE MATERIAL DURING THE SITTING.

[Answer ALL Questions]

[Use a Separate Answer Book for THIS SECTION]

Section 2:

2. Many different models of software engineering have been proposed.

   (a) Briefly describe two models of software engineering and state the advantages and disadvantages of each. [6 %]

   (b) How are the costs of software development distributed across the lifecycle of a typical project? Explain the reason(s) for this distribution of costs. [2 %]

3. Requirements are the basis of all software development.

   (a) Give two reasons why it can be difficult to develop a requirements specification for a project. Briefly describe one way of attempting to overcome these difficulties. [3 %]

   (b) Briefly explain the difference between *functional* and *non-functional* requirements. [2 %]

   (c) List four characteristics you would expect a completed requirements specification to have and briefly explain why they are important. [4 %]

4. *Function-oriented* design and *Object-oriented* design are two widely used structured design methods.

   (a) Briefly explain the difference between the two methods. For each method, give an example of a type of problem for which it is most appropriate. Give reasons for your choice. [3 %]

   (b) For each method, state two types of system model developed during the design process and briefly explain their role in design. [4 %]

   (c) Briefly outline the advantages and disadvantages of each method. [4 %]

THIS IS NOT AN OPEN-BOOK EXAMINATION - CANDIDATES MAY NOT CONSULT ANY REFERENCE MATERIAL DURING THE SITTING.

5. (a) What is the difference between *verification* and *validation*? [2%]

(b) Explain the difference between static verification and testing. State two different static verification techniques, and, for each, identify the objective of the testing. [4%]

THIS IS NOT AN OPEN-BOOK EXAMINATION - CANDIDATES MAY NOT CONSULT ANY REFERENCE MATERIAL DURING THE SITTING.

[Answer ALL Questions]

[Use a Separate Answer Book for THIS SECTION]

Section 3

6. Assume that the list L contains [2, 6, 8, 7] (2 is the 0-element, 6 is the 1-element, etc.). What does L contain after the following sequence of statements has been executed? Justify your answer.

```
    int s = L.size-1;
for (int I = 0; I <= s; I++)
   { L.insert(L.retrieve(s - I )*2, I);}
```

[2%]

THIS IS NOT AN OPEN-BOOK EXAMINATION - CANDIDATES MAY NOT CONSULT ANY REFERENCE MATERIAL DURING THE SITTING.

7. Consider the following modification of the linked list design we saw in class:

```
   class DumLinkedList
{ private class Node
    { char item;
      Node next;

      Node (char c, Node ptr)        // constructor of Node
        { item = c; next = ptr;}
      }

  private Node first;

  DumLinkedList()         // constructor of DumLinkedList
    { first = new Node('n',null);}

  public void cons(char newitem)
    { first.next = new Node(newitem, first.next);}

  public boolean insert(char newitem, int index)
  // insert newitem at index position
     {…}
}
```

This linked list design differs from the one we saw in class in that first points to a *dummy* Node component which in turns points to the first element of the list, instead of First pointing directly to the first element. The content of the item field of the *dummy* is non-important (I have chosen to initialize it with 'n', but any other character can go there as well), because *dummy* is NOT an element of the list.

(a) Draw two pictures to visualize the results of each of the following statements (the code for the constructor and for the cons method are provided in the above class definition):

```
DumLinkedList L = new DumLinkedList();
L.cons('a');
```

[3%]

(b) Write the method insert for the class DumLinkedList.

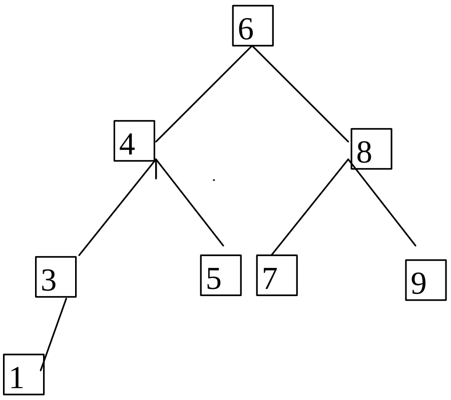
[3%]

(c) What is the main (positive) consequence of having a *dummy* component?

[1%]

THIS IS NOT AN OPEN-BOOK EXAMINATION - CANDIDATES MAY NOT CONSULT ANY REFERENCE MATERIAL DURING THE SITTING.

8. Consider the following binary tree:



(a) Name the root element, its right and left children, and list the leaf elements.

[2%]

(b) What's the size of the tree? What's its height?

[2%]

(c) List the elements using the 'postorder' traversal.

[2%]

(d) Is the tree an AVL tree? Is the tree a heap? Justify your answers.

[3%]

9. Show step-by-step the effect of inserting the elements 10, 30, 50, 20, 70, 60 into an initially empty AVL tree.

[4%]

10. Propose an appropriate data structure for each of the following tasks. Justify your choice in terms of efficiency of the main operations required, if appropriate.

(a) We want to represent a library catalogue. Each item to be stored in the catalogue consists of a description and an 8-digit code with uniquely identifies a book. The maximum possible number of books in the catalogue is approximately 10^8. The main operations to be performed will be to search for a book using the 8-digit code as key, insert, and also print the catalogue following an ascending order of the codes.

[3%]

(b) A data structure is required to store information about the planned underground railway for a city. All stations, together with the connections and the timetable information, should be stored. The data structure will be the basis for passengers enquiries such as “What is the quickest way to get from station A to station B?”

[3%]

11. For each of the following functions *f*, where n = 1, 2, ,3,… find the smallest function *g* in the order hierarchy such that *f* is O(*g*):

(a) $f(n) = (2 + n) * (3 + \log(n))$

(b) $f(n) = n *(3 + n) - 7*n$

(c) $f(n) = \log(n^2) + n$

[3%]

12. Let A be an array of N integers, indexed from 1 to N. What is the time complexity of the following program fragment? Justify your answer. (Hint: best, worst and average complexity are all the same).

```
int pos; int temp;
for (int I=1;I<=N-1;I++)
 { pos = I;
   for (int J=I+1;J<=N;J++)
     if (A[j]<A[pos]) pos=J;
   temp=A[I]; A[I]=A[pos]; A[pos]=temp;
 }
```

[3%]