3 HOURS

ANSWER ALL 4 QUESTIONS

Electronic calculators are not permitted.

Question 1

Answer any *two* parts from (a), (b), (c), (d):

Note: where an algorithm is asked for, you may write in any suitable pseudo-code. Correct syntax for any computer language is not expected.

Part (a)

i) Give a definition for a **sequence** in terms of the values it can hold and the operations that can be performed on it. [2 marks]

ii) A new Abstract Data Type is proposed in which elements of a list can be inserted and deleted at the two ends but nowhere else. Using this definition, define the public interface for a class to implement this ADT [2 marks]

iii) It is suggested to implement the ADT of part (ii) using a privately inherited sequence. Show how to implement the operations defined in part (ii) in terms of the public operations of a sequence. [2 marks]

iv) Discuss the relative efficiency of the implementation in part (iii) if the underlying sequence is implemented in terms of

1) an array
2) a linked list
3) a double linked list

In each case, discuss the modifications you would make to improve efficiency if using these structures for a direct implementation, rather than through the private inheritance mechanism. [6 marks]

QUESTION 1 CONTINUES OVER PAGE

TURN OVER

Part (b) An *ordered_queue* is an Abstract Data Type defined as a set of elements <e_1, e_2,...e_n> with the property that $e_i \le e_j$ whenever $i < j$, together with the operations **enqueue**(element), **dequeue**(), and **front**().

i) Describe using diagrams, or pseudo-code, or otherwise, how to implement this ADT using a linked list. [4 marks]

ii) Would there be any advantage in using a circular linked list ? Give your reasons. [2 marks]

iii) It is required to have another operation **merge**(ordered_queue) that takes all the elements from one ordered_queue and placing them in a second, emptying the first ordered_queue in the process. Show that this operation can be implemented using only the three operations already defined. [3 marks]

iv) Show that the merge operation can be made more efficient by adding another function **next**() that returns the value of the element following the last one inserted. [3 marks]

Part (c) You are writing a simple computer game in which the hero has to open a series of doors, behind each of which is a key. Some doors have no lock, whilst others have more than one lock. Each key is unique. The object of the game is to open a final door to get some treasure.

i) Show how to re-express this problem in terms of a directed graph, identifying the set corresponding to the vertices, the relationship corresponding to the edges, and the function of the graph corresponding to the desired result. [3 marks]

ii) Describe an efficient algorithm for solving the problem. Indicate what data structures you might employ, if any, and suggest a suitable representation for the graph. [9 marks]

QUESTION 1 CONTINUES ON NEXT PAGE

Part (d)

i) Define a binary tree, and a binary search tree. [2 marks]

ii) If a binary tree has n internal nodes, what are the minimum and the maximum values for the total height of the tree ? [2 marks]

iii) Show the binary search tree resulting from the insertion of the following names, in the order given, and state its height:

Ruth, Chris, Brian, Anne, Zak, Wanda, Tom

[2 marks]

iv) It is proposed to modify the method of insertion in a binary search tree in the following way. After insertion, traverse the tree in order until the median node is reached, then insert this node in a new tree. Repeat, recursively for the first half of the nodes in the original tree, then for the second half. Finally replace the original tree with the new tree.

Express the above algorithm in pseudo-code, taking care to terminate recursion correctly and to account for all nodes in the original tree once and once only. Give the resulting tree for the example of part (iii) [4 marks]

v) Discuss whether the algorithm of part (iv) is a reasonable one to reduce the complexity of binary search. What problem will occur when deletion is also allowed on the tree ? [2 marks]

END OF QUESTION 1

Question 2

Answer part (a) and either part (b) or part (c).

Part (a) Let $A \sim B$ mean that A and B are poly-time equivalent to each other.

(i) Define what it means when we say that two decision problems are poly-time equivalent. [4 marks]

(ii) Prove that ~ is an *equivalence relation* , i.e. reflexive, symmetric and transitive [8 marks]

Part (b) (i) Define the following classes:-

1. **P**
2. **NP**
3. **NPC** [6 marks]

(ii) Prove that **NPC** is closed under poly-time equivalence, i.e. if $A \in$ **NPC** and $A \sim B$ then $B \in$ **NPC** [7 marks]

Part (c)

(i) Explain what the *halting problem* is. [5 marks]

(ii) A *pair of primes (p, p+2)* consists of two positive integers that differ by two, such that both p and $p+2$ are prime numbers. Examples of pairs of primes are (3, 5), (11, 13) and (29, 31).

Suppose that the halting problem were decidable and that there were a Turing machine H that could solve it. Let P be a Turing machine that takes two positive integers *(m, n)* as its input and returns 'yes' if *(m, n)* is a pair of primes and 'no' otherwise. Suppose P is given to you (i.e. you don't have to define this).

Use H and P to define a procedure Π that takes an integer n as its input and returns 'yes' if there is a prime number $p \geq n$ such that *(p, p+2)* is a pair of primes and it returns 'no' otherwise. You may define Π using pseudo-code, by defining a Turing machine for Π, or any way you like, provided it is clear. [4 marks]

(iii) Now use your procedure Π, from the previous part, and H to devise a procedure G that takes no input at all, but returns 'yes' if there are infinitely many pairs of primes and 'no' otherwise. [4 marks]

END OF QUESTION 2

Question 3

Answer All parts, both a and b.

a)

i) Draw an annotated syntax diagram for an assignment of the form:-

ID = expression ; [3 marks]

ii) Draw annotated syntax diagrams defining **expression**.

Your definition should involve operands (tokenised as NUM and ID), the relational operators (tokenised as RELOP), arithmetic operators (tokenised as ADDOP and MULOP), and parentheses. Your syntax diagrams should give MULOP the highest priority, followed by ADDOP then RELOP. You may allow expressions such as

(x < y) < z

but not

x < y < z.

[5 marks]

b)

In the C programming language a **conditional expression** is one which uses the conditional operator ? and takes the form

exp ? exp1 : exp2

If *exp* when evaluated is true (non-zero), the result is *exp1*, otherwise it is *exp2*. For example, *x<2 ? 3 : 4* evaluates to *3* when *x<2* and *4* otherwise.

i) Show the kind of code which you would expect to be compiled for the following assignment

z = x < y ? x : y;

[4 marks]

ii) Extend your definition of **expression** from part (a) to include **conditional expression**. You need not consider nested conditional expressions. You may rename the existing expression syntax diagram.

[5 marks]

iii) Annotate your modified syntax diagram to show how code as in your answer to part (i) of this section could be generated.

[2 marks]

iv) Discuss whether a conditional expression could contain expressions which are themselves conditional expressions. Consider questions such as the following (but do not limit yourself to them): Does it make sense? (consider some examples). How could your syntax diagrams be modified to allow it?

[6 marks]

END OF QUESTION 3

Question 4

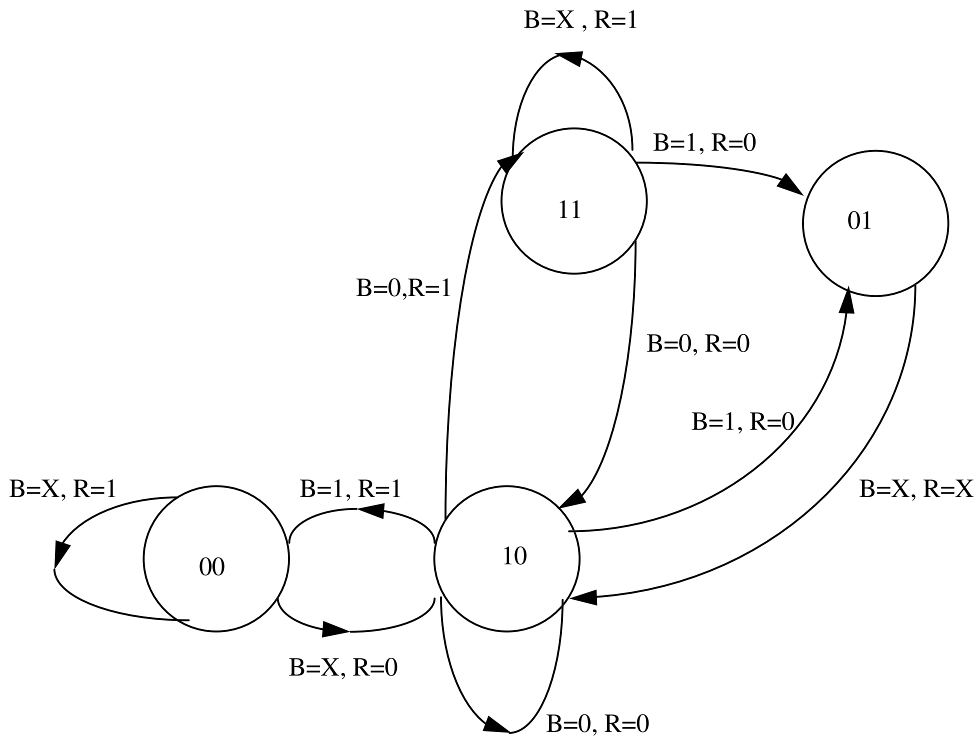
D0B 1997

Answer part (a) and 2 parts from parts (b)-(e):

(a) What is the role of transistors in digital electronic circuits? [3 Marks]

Explain the difference between p-type and n-type semi-conductors, and explain the action of a *p-n junction.* [4 Marks]

Describe the structure and operation of either a p-n-p or a n-p-n transistor. [6 Marks]



(b) Draw the truth table for the Finite State Machine (FSM) with the following State Diagram:

Use Karnaugh maps or other technique to derive the equations for the next state bits.

Draw the final circuit for the Finite State Machine using AND, OR and INVERTER gates, and flip-flops for the state store. [6 Marks]

(c) The R-S flip-flop is a 1-bit storage device.

Draw a logic diagram for such device and give its truth table.

What is the basic property of the device that makes it a binary storage device and what are the roles of the R and S inputs? [6 Marks]

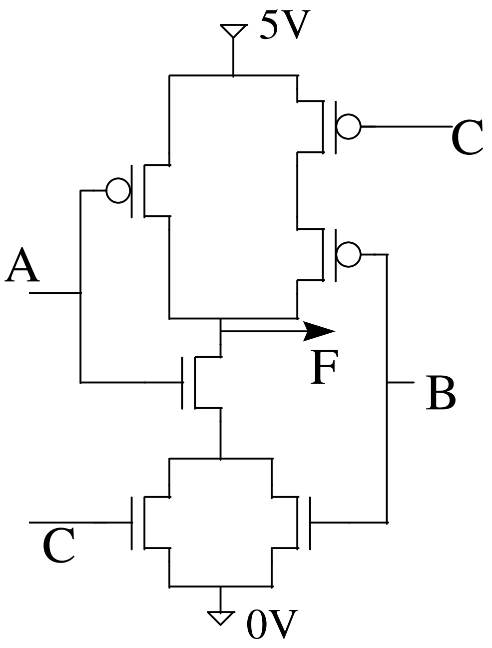
CONTINUED

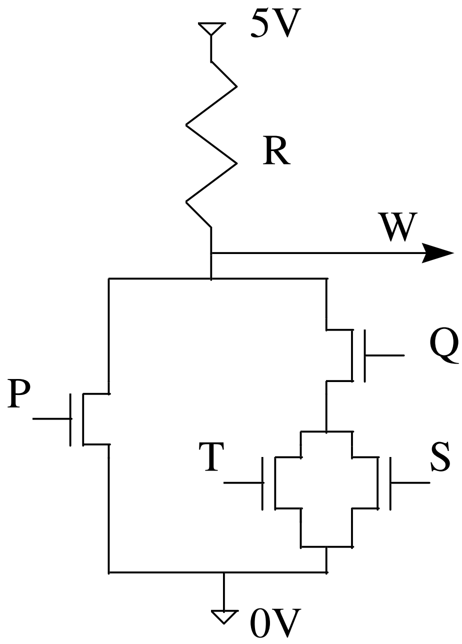
QUESTION 4 CONTINUES ON NEXT PAGE

(d) The datapath inside a modern CPU usually has a 3-bus architecture rather than a 1-bus or 2-bus architecture.

Explain the use of these busses by the datapath and explain why a 3-bus architecture is now used. [6 Marks]

(e) Using truth tables and Karnaugh maps or other means to derive the logical function performed by both the following circuits:-





[6 Marks]

END OF QUESTION 4

END OF PAPER