Oxford Cambridge and RSA

Monday 19 June 2023 – Morning

A Level Computer Science

H446/02 Algorithms and programming

Time allowed: 2 hours 30 minutes

You can use:
- a ruler (cm/mm)
- an HB pencil

Do not use:
- a calculator

Please write clearly in black ink. **Do not write in the barcodes.**

Centre number						Candidate number				

First name(s) ____________________

Last name ____________________

INSTRUCTIONS
- Use black ink. You can use an HB pencil, but only for graphs and diagrams.
- Write your answer to each question in the space provided. If you need extra space use the lined pages at the end of this booklet. The question numbers must be clearly shown.
- Answer **all** the questions.

INFORMATION
- The total mark for this paper is **140**.
- The marks for each question are shown in brackets **[ ]**.
- Quality of extended response will be assessed in questions marked with an asterisk (*).
- This document has **32** pages.

ADVICE
- Read each question carefully before you start your answer.

BLANK PAGE

PLEASE DO NOT WRITE ON THIS PAGE

Section A

1 A tree is one example of a data structure.

(a) **(i)** Give **two** characteristics of a tree data structure.

1 ....

....

2 ....

.... **[2]**

(ii) The following data is entered into a binary search tree.

22 13 5 36 55 14 8

Draw the binary search tree when the given data is entered in the order given.

[4]

(iii) Describe how a **leaf node** is deleted from a binary search tree.

....

....

....

.... **[2]**

(iv) Describe how a binary search tree can be searched for a value.

....

....

....

....

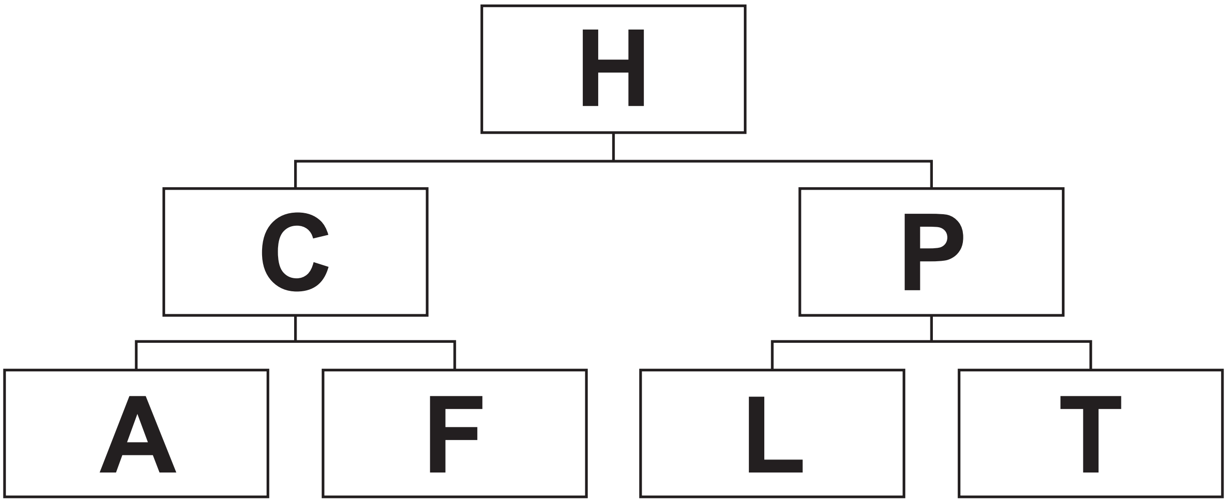
....

....

....

.... **[4]**

(v) Identify the order that the nodes will be visited in a **depth-first (post-order)** traversal of this binary search tree.



.... **[4]**

(vi) Explain how backtracking is used in depth-first (post-order) traversals.

....

....

....

.... **[2]**

(b) A graph is another type of data structure.

An example graph is shown in **Fig. 1**.

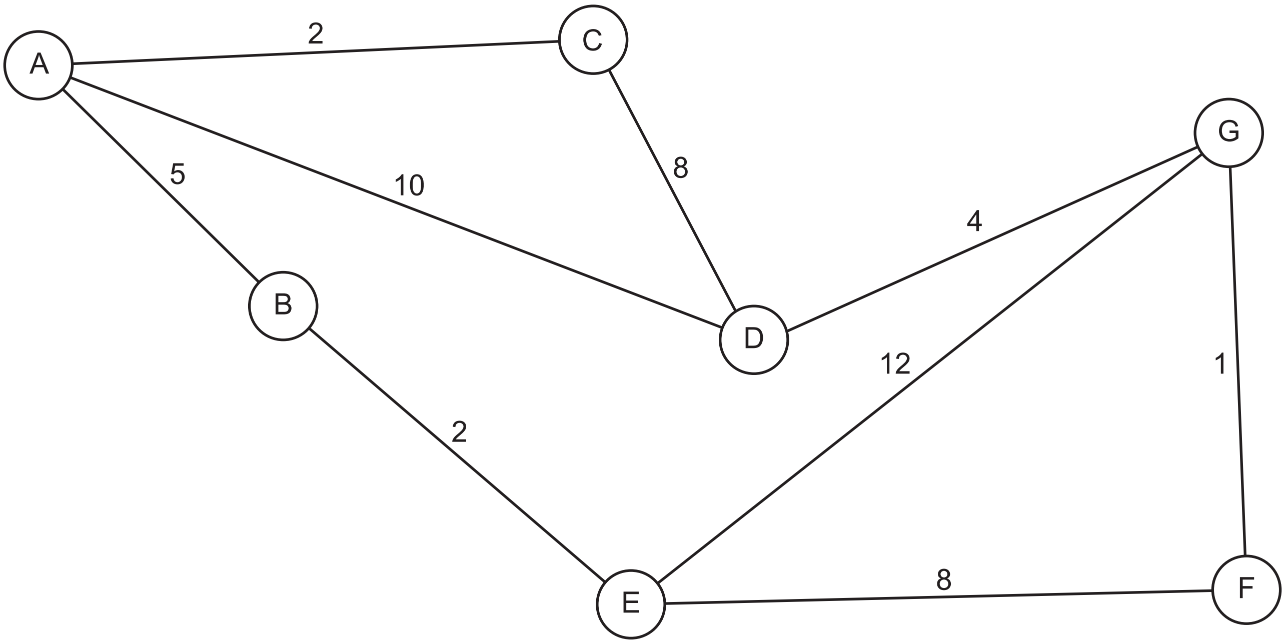


Fig. 1

Show how Dijkstra's algorithm can be used on the graph shown in **Fig. 1** to find the shortest path from start node A to end node G.

You must state the nodes on the final path and the distance of this path. Show your working.

You may use the table below to give your answer.

....

....

....

....

....

....

....

....

....

....

....

....

Node	**Distance travelled**	**Previous node**

Final path: ....

Distance: ....

[6]

2* A company needs a new computer program that will create schedules for delivery drivers. It will need to identify a possible order that the drivers can deliver items and possible routes they could take.

Discuss how programmers could make use of problem recognition and problem decomposition when designing this system.

You should include the following in your answer:

- a description of both problem recognition and decomposition
- how each method can be used when designing the solution
- the benefits of using each method when designing the solution.

[9]

....

....

....

....

....

....

3 A program stores data in a linked list.

The current contents of the linked list are shown in **Fig. 3**, along with the linked list pointers.

headPointer	1
freeListPointer	4

location	data	pointer
0	"blue"	6
1	"red"	0
2	"green"	8
3	"orange"	NULL
4		5
5		7
6	"grey"	2
7		9
8	"purple"	3
9		NULL

Fig. 3

(a) State the purpose of `headPointer` and `freeListPointer` in the linked list shown in **Fig. 3**.

headPointer ....

....

freeListPointer ....

....

[2]

(b) State the meaning of the pointers with the value `NULL` in the linked list shown in **Fig. 3**.

....

.... **[1]**

(c) A procedure outputs the data in the linked list shown in **Fig. 3** from the first item in the list, to the last item.

Give the output from the procedure.

....

.... **[2]**

(d) A new item needs to be added to the linked list.

Describe how a new item is added to a linked list.

....

....

....

....

....

....

....

.... **[4]**

(e) The function `findNode` will search the linked list and return either the position of the node that contains the data item, or -1 if the data item is not found.

The data held in a node at location `x` can be accessed with `linkedList[x].data`. The pointer of the node at location `x` can be accessed with `linkedList[x].pointer`.

For example, using the linked list shown in **Fig. 3**:
`linkedList[2].data` returns `green`.
`linkedList[2].pointer` returns `8`.

Complete the function, using pseudocode or program code.

```
function findNode(toFind, headPointer, linkedList)
  currentNode = ..................................
  while(currentNode != ..................................)
    if linkedList[currentNode]. .................................. == toFind then
      return currentNode
    else
      currentNode = linkedList[..................................].pointer
    endif
  endwhile
  return ..................................
endfunction
```

[5]

4 A programmer has designed a program that includes a reusable program component.

(a) The reusable program component is a function called `isInteger()`. This will take a string as an argument and then check that each digit is between 0 and 9. For example if 103 is input, it will check that the digits 1, 0 and 3 are each between 0 and 9.

The `asc()` function returns the ASCII value of each digit. For example `asc("1")` returns 49.

The ASCII value for 0 is 48. The ASCII value for 9 is 57.

```
01   function isInteger(number)
02     result = true
03     for count = 0 to number.length-1
04       asciiValue = asc(number.substring(count, 1))
05       if not(asciiValue >= 48 and asciiValue <= 57) then
06         result = false
07       endif
08     next count
09     return result
10   endfunction
```

(i) Identify **one** identifier used in the function `isInteger()`.

.... **[1]**

(ii) Give the line number where the branching (selection) construct starts in the function `isInteger()`.

.... **[1]**

(iii) Give the line number where the iteration construct starts in the function `isInteger()`.

.... **[1]**

(b) Describe the purpose of the following lines in the function `isInteger()`.

Line 03 ....

....

Line 04 ....

....

Line 09 ....

....

[3]

(c) Give **two** reasons why reusable program components are used in programs.

1 ....

....

2 ....

....

[2]

5 A recursive pseudocode function, `recursiveAlgorithm()`, is shown.

```
function recursiveAlgorithm(value)
  if value <= 0 then
    return 1
  elseif value MOD 2 = 0 then
    return value + recursiveAlgorithm(value – 3)
  else
    return value + recursiveAlgorithm(value – 1)
  endif
endfunction
```

(a) Describe the key features of a recursive algorithm.

You may refer to the function, `recursiveAlgorithm()` in your answer.

....

....

....

....

....

.... **[3]**

(b) Trace the recursive function, `recursiveAlgorithm()`, and give the final return value when called with `recursiveAlgorithm(10)`. You may choose to use the table below to give your answer.

....

....

....

....

....

....

....

....

....

....

....

....

....

Function call	`value`	`return`

Final return value .... **[5]**

6 Octal is a base 8 number system.

To convert a denary number to base 8:

- the denary value is divided by 8 and the remainder is stored
- the integer value after division is divided by 8 repeatedly until 0 is reached
- the remainders are then displayed in reverse order.

Example 1:

Denary 38

```
        38 / 8 = 4 remainder 6      6
        4 / 8 = 0 remainder 4       4
```

Octal = 46

Example 2:

Denary 57

```
        57 / 8 = 7 remainder 1      1
        7 / 8 = 0 remainder 7       7
```

Octal = 71

Write an algorithm to:

- take a denary value as input from the user
- convert the number to octal
- output the octal value.

You do **not** need to validate the input from the user.

Write your algorithm using pseudocode or program code.

....

....

....

....

....

....

....

....

....

....

....

....

.... **[6]**

BLANK PAGE

PLEASE DO NOT WRITE ON THIS PAGE

7* **(a)** A program designer needs to decide on an algorithm to use from a choice of three. The table shows the worst-case Big O complexities for each algorithm.

Algorithm	**Time Complexity**	**Space Complexity**
1	Linear	Exponential
2	Exponential	Constant
3	Logarithmic	Logarithmic

The program will be used to analyse data that can range from 2 items to 2 billion items.

Compare the use of all **three** algorithms and suggest which the programmer should use.

You should include the following in your answer:

- the meaning of constant, logarithmic, linear and exponential complexity
- how well each algorithm scales as the amount of data increases
- which algorithm is the most suitable for the given task.

[9]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

(b) The program designer is investigating the use of concurrent processing.

(i) Describe what is meant by the term ‘concurrent processing’.

....

....

....

.... **[2]**

(ii) Give **two** benefits of using concurrent processing.

1 ....

....

2 ....

....

[2]

(c) The programmer needs to use a merge sort in one part of the problem to sort items in ascending order.

(i) Describe how a merge sort works.

....

....

....

....

....

....

....

....

....

.... **[5]**

(ii) Give **one** benefit and **one** drawback of the programmer using a merge sort instead of a bubble sort.

Benefit ....

....

Drawback ....

....

[2]

(d) The programmer uses an Integrated Development Environment (IDE).

Complete the table by identifying **and** describing **three** IDE features that can help the programmer to develop, or debug a program.

IDE feature	Description

[6]

8 A program is being designed that will allow a user to log into an account on a website using a username and password.

(a) Identify **two** possible inputs and **one** output this program will need.

Input 1 ....

....

Input 2 ....

....

Output ....

....
[3]

(b) Identify **two** possible sub-procedures that could be used in this program.

1 ....

....

2 ....

....
[2]

BLANK PAGE

PLEASE DO NOT WRITE ON THIS PAGE

Section B

9 A text-based computer game allows a user to dig for treasure on an island. The island is designed as a grid with 10 rows and 20 columns to store the treasure. Each square is given an x and y coordinate. Some of the squares in the grid store the name of a treasure object. Each treasure object has a value, e.g. 100 and a level, e.g. "Bronze."

(a) The computer game makes use of abstraction.

(i) Describe what is meant by the term abstraction and give an example of how abstraction can be used in the treasure game.

Description: ....

....

....

....

Example: ....

.... **[3]**

(ii) Give **three** benefits of using abstraction when writing a program.

1 ....

....

2 ....

....

3 ....

.... **[3]**

(b) The treasure game is being programmed using an object-oriented paradigm.

A class, `Treasure`, is used to store the treasure objects.

The design for the `Treasure` class, its attributes and methods is shown here.

```
class: Treasure
--------------------------------
attributes:
private value : integer
private level : string
--------------------------------
methods:
new()
function getValue()
function getLevel()
```

(i) The constructor method takes a value as an integer, e.g. 100, and a level, e.g. "bronze", as parameters and assigns these to the attributes.

Write pseudocode or program code to declare the class `Treasure`.

You should define the attributes and constructor method in your answer.

You do **not** need to write the get methods.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... **[5]**

(ii) The get method `getLevel()` will return the appropriate attribute.

Write the method `getLevel()` using either pseudocode or program code.

....

....

....

.... **[2]**

(iii) Describe the object-oriented programming technique being used in part **9(b)(ii)**.

....

....

....

.... **[2]**

(c) A class, `Board`, is used to store the 10 row (x coordinate) by 20 column (y coordinate) grid.

The design for the `Board` class, its attributes and methods is shown here.

```
class: Board
----------------------------------------------
attributes:
private grid : Array of Treasure
----------------------------------------------
methods:
new()
function getGridItem(x, y)
function setGridItem(x, y, treasureToInsert)
```

The constructor initialises each space in the grid to a treasure object with `value` as -1 and `level` as an empty string.

Complete the following pseudocode for the constructor method.

```
public procedure new()
  for row = ............................ to 9
    for column = 0 to ............................
      ............................ [row, column] = new Treasure(............................,"")
    next ............................
  next row
endprocedure
```

[5]

(d) A procedure, `guessGrid()`:

- takes a `Board` object as a parameter
- accepts the row (x) and column (y) coordinates from the user
- outputs "No treasure" if there is no treasure found at the coordinate (level is an empty string)
- if there is treasure at that coordinate, it outputs the level and the value of the treasure in an appropriate message.

Write the procedure `guessGrid()` using either pseudocode or program code.

....

....

....

....

....

....

....

....

....

....

....

....

....

.... **[7]**

(e) Describe **two** benefits of using an object-oriented paradigm rather than a procedural paradigm.

1 ....

....

....

....

2 ....

....

....

....

[4]

(f)* The main program initialises a new instance of `Board`. The programmer is considering declaring this as a global variable or as a local variable and then passing this into the subroutines that control the game.

Compare the use of variables and parameters in this game.

You should include the following in your answer:

- what is meant by a local variable and global variable
- how local and global variables can be used in this program
- the use of passing parameters by value and by reference.

[9]

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

END OF QUESTION PAPER

ADDITIONAL ANSWER SPACE

If additional space is required, you should use the following lined page(s). The question number(s) must be clearly shown in the margin(s).

OCR
Oxford Cambridge and RSA