3.4 Searching algorithms

Name:	
Class:	
Date:	

Time:	**250 minutes**
Marks:	**162 marks**
Comments:	

Q1.

(a) This question refers to the subroutine `CreateTileDictionary`.

The points values for the letters J and X are to be changed so that they are not worth as many points as the letters Z and Q.

Adapt the subroutine `CreateTileDictionary` so that the letters J and X are worth 4 points instead of 5 points.

Test that the changes you have made work:

- run the **Skeleton Program**
- enter `2` at the main menu
- enter `1` to view the letter values.

Evidence that you need to provide

(i) Your PROGRAM SOURCE CODE for the amended subroutine `CreateTileDictionary`.

(1)

(ii) SCREEN CAPTURE(S) showing the results of the requested test.

(1)

(b) This question refers to the subroutine `Main`.

Currently each player starts with 15 letter tiles. In the `Main` subroutine `StartHandSize` is set to a value of 15.

Change the subroutine `Main` so that the user can choose what the value for `StartHandSize` will be.

Before the main menu is displayed and before the first iteration structure in `Main` the message "`Enter start hand size:` " should be displayed and the user's input should be stored in `StartHandSize`. This should happen repeatedly until the user enters a value for the start hand size that is between 1 and 20 inclusive.

Test that the changes you have made work:

- run the **Skeleton Program**
- enter `0` when asked to enter the start hand size
- enter `21` when asked to enter the start hand size
- enter `5` when asked to enter the start hand size
- enter `1` at the main menu.

Evidence that you need to provide

(i) Your PROGRAM SOURCE CODE for the amended subroutine `Main`.

(4)

(ii) SCREEN CAPTURE(S) showing the requested test. You must make sure that evidence for all parts of the requested test is provided in the SCREEN CAPTURE(S).

(1)

(c) This question refers to the subroutine `CheckWordIsValid`.

When a player enters a word, a linear search algorithm is used to check to see if the word entered is in the list of `AllowedWords`. The subroutine `CheckWordIsValid` is to be changed so that it uses a more time-efficient search algorithm.

Change the `CheckWordIsValid` subroutine so that it uses a binary search algorithm instead of a linear search algorithm.

You **must write your own search routine** and not use any built-in search function that might be available in the programming language you are using.

Each item in `AllowedWords` that is compared to the word that the user entered should be displayed on the screen.

Figure 2 shows examples of how the new version of `CheckWordIsValid` should work if `AllowedWords` contained the items shown in **Figure 1**.

Figure 1

BIG
BUG
FED
GET
JET
NOT
SIP
WON

Figure 2

1) If the user enters the word BIG then a value of `True` should be returned and the words GET, BUG, BIG should be displayed, in that order.
2) If the user enters the word JET then a value of `True` should be returned and the words GET, NOT, JET should be displayed, in that order.
3) If the user enters the word ZOO then a value of `False` should be returned and the words GET, NOT, SIP, WON should be displayed, in that order.

Test that the changes you have made work:

- run the **Skeleton Program**
- if you have answered part (b), enter `15` when asked to enter the start hand size; if you have not answered part (b) yet then skip this step
- enter `2` at the main menu
- enter the word `jars`.

Evidence that you need to provide

(i) Your PROGRAM SOURCE CODE for the amended subroutine `CheckWordIsValid`.

(8)

(ii) SCREEN CAPTURE(S) showing the requested test.

(1)

(d) This question extends the functionality of the game.

After spelling a valid word the player decides which one of four options to select to determine how many tiles will be added to their hand. Before choosing they can look at the values of the tiles.

It would help the player make their decision if they were aware of how useful each letter was by knowing the frequency with which each letter appears in the list of allowed words.

The program is to be extended so that when the player chooses to view the tile values they are also shown the number of times that each letter appears in the list of allowed words.

What you need to do

Task 1
Create a new subroutine called `CalculateFrequencies` that looks through the list of allowed words and displays each of the 26 letters in the alphabet along with the number of times that the letter appears in the list of allowed words, which the subroutine has calculated.

Task 2

Modify the `DisplayTileValues` subroutine so that after displaying the tile values it also calls the `CalculateFrequencies` subroutine.

Task 3
Test that the changes you have made work:

- run the **Skeleton Program**
- if you have answered part (b), enter `15` when asked to enter the start hand size; if you have not answered part (b) yet then skip this step
- enter `2` at the main menu
- enter `1` to display the letter values.

Evidence that you need to provide

(i) Your PROGRAM SOURCE CODE for the new subroutine `CalculateFrequencies`, the amended subroutine `DisplayTileValues` and any other subroutines you have modified when answering this question.

(8)

(ii) SCREEN CAPTURE(S) showing the requested text.

(1)

(e) The scoring system for the game is to be changed so that if a player spells a valid word they score points for all valid words that are a prefix of the word entered. A prefix is the first **x** characters of a word, where **x** is a whole number between one and the number of characters in the word.

In the **Skeleton Program**, `AllowedWords` contains the list of valid words that have been read in from the **Data File aqawords.txt**.

Example

If the user enters the word TOO they will be awarded points for the valid words TOO and TO as TO is a prefix of the word TOO. They will not be awarded points for the word OO even though it is a valid word and is a substring of TOO because it is not a prefix of TOO.

Example
If the user enters the word BETTER they will be awarded points for the words BETTER, BET and BE as these are all valid prefixes of the word entered by the user. They would not be awarded points for BETT or BETTE as these are not valid English words. They would not be awarded points for BEER as even though it is contained in the word BETTER it is not a prefix.

Example
If the user enters the word BIOGASSES they will be awarded points for the words BIOGASSES, BIOGAS, BIOG, BIO and BI as these are all valid prefixes of the word entered by the user. They would not be awarded points for BIOGA, BIOGASS or BIOGASSE as these are not valid English words. They would not be awarded points for GAS as even though it is contained in the word BIOGASSES it is not a prefix.

Example
If the user enters the word CALMIEST they will not be awarded any points as even though CALM at the start is a valid word the original word entered by the user, CALMIEST, is not.

Example
If the user enters the word AN they will be awarded points for the word AN. They would not be awarded points for A even though A at the start is a valid word as points are only awarded for words that are at least two letters long.

What you need to do

Task 1
Write a **recursive** subroutine called `GetScoreForWordAndPrefix` that, if given a valid word, returns the score of the word added to the score for any valid words that are prefixes of the word.

To get full marks for this task the `GetScoreForWordAndPrefix` subroutine must make use of recursion in an appropriate way to calculate the score for any prefixes that are also valid words.

If your solution uses an alternative method to recursion you will be able to get most but not all of the available marks for this question.

Task 2
Modify the `UpdateAfterAllowedWord` subroutine so that it calls the new `GetScoreForWordAndPrefix` subroutine instead of the `GetScoreForWord` subroutine.

Task 3
Test that the changes you have made to the program work:

- run the **Skeleton Program**
- if you have answered part (b), enter `15` when asked to enter the start hand size; if you have not answered part (b) yet then skip this step
- enter `2` at the main menu
- enter the word `abandon`
- enter `4` so that no tiles are replaced.

Evidence that you need to provide

(i) Your PROGRAM SOURCE CODE for the new subroutine `GetScoreForWordAndPrefix`, the amended subroutine `UpdateAfterAllowedWord` and any other subroutines you have modified when answering this question.

(11)

(ii) SCREEN CAPTURE(S) showing the results of the requested test.

(1)
(Total 37 marks)

Q2.

The table below lists some well-known algorithms.

Algorithm
Linear search
Merge sort
Binary search
Post-order tree-traversal

(a) Which of the algorithms listed in the table has $O(n \log n)$ time complexity?

____

____

(1)

(b) How many of the algorithms listed in the table are algorithms used to solve tractable problems?

____

____

(1)
(Total 2 marks)

Q3.

The code below shows an incomplete algorithm for a binary search.

```
PROCEDURE BSearch(List, F, L,
ItemToFind)
  Found ← False
  Failed ← (1)__________________
  WHILE NOT Failed AND NOT Found
    M ← (F + L) DIV 2
    IF List[M] = ItemToFind
```

```
          THEN Found ← True
          ELSE
            IF F >= L
              (2)__________________
              ELSE
                IF List[M] > ItemToFind
                  THEN (3)__________________
                  ELSE F ← M + 1
                ENDIF
            ENDIF
        ENDIF
      ENDWHILE
      IF Found = True
        THEN OUTPUT "Item is in list"
        ELSE OUTPUT "Item is not in list"
      ENDIF
    ENDPROCEDURE
```

The `DIV` operator calculates the whole number result of integer division. For example, `15 DIV 4 = 3, 17 DIV 4 = 4`.

(a) What code should be added at position **(1)**?

____

____

(1)

(b) What code should be added at position **(2)**?

____

____

(1)

(c) What code should be added at position **(3)**?

____

____

____

____

(2)

The table below contains a list of orders of time complexity (in no particular order).

Order of time complexity
O(1)
O(n^2)
O(log n)

$O(k^n)$
$O(n)$

Which of the orders of time complexity given in the table :

(d) could be the time complexity of an intractable problem?

____

____

(1)

(e) is the time complexity for a binary search?

____

____

(1)

(f) is the time complexity for getting the first item in a list?

____

____

(1)

(g) is the time complexity for a linear-search algorithm?

____

____

(1)

(h) Explain why a linear-search has the order of time complexity given in your answer to question (g).

____

____

____

____

(2)

(Total 10 marks)

Q4.

The binary search method can be used to search for an item in an ordered list.

(a) Show how the binary search method works by writing numbers in the table below to indicate which values would be examined to determine if the name “Richard” appears in the list.

Write the number “1” by the first value to be examined, “2” by the second value to be examined and so on.

Position	Value	Order Examined In
1	Adam	
2	Alex	
3	Anna	
4	Hon	
5	Mohammed	
6	Moonis	
7	Niraj	
8	Philip	
9	Punit	
10	Ravi	
11	Richard	
12	Timothy	
13	Tushara	
14	Uzair	
15	Zara	

(3)

(b) A different list contains 137 names.

What is the maximum number of names that would need to be accessed to determine if the name “Rachel” appears in the list? Write your answer in the box below.

$\boxed{}$

(1)

(c) Tick **one** box to indicate the order of time complexity of the binary search method.

Order of time complexity	Tick one box
$O(\log_2 n)$	
$O(n)$	

O(n^2)	

(1)
(Total 5 marks)

Q5.

A *recursively-defined* procedure **ProcA** that takes two integers as parameters is defined below.

(a) What is meant by a recursively-defined procedure?

____

____

(1)

(b) What is the role of the stack when a recursively-defined procedure is executed?

____

____

(1)

(c) Dry run the procedure call **ProcA(11,1)** using the data in the array, **Items**, by completing the trace table below.

```
Procedure ProcA (Number,Entry)
      If Number <> Items[Entry]
      Then ProcA (Number,Entry+1)
      Else Output (Entry)
      EndIf
EndProc
```

	Items
[1]	4
[2]	5
[3]	8
[4]	11
[5]	15
[6]	19
[7]	21
[8]	28
[9]	33

Number	Entry	Output
11	1	

(4)

(d) What is the purpose of this algorithm?

____

(1)

(e) Give a situation where this algorithm will fail.

____

____

(1)

(f) Suggest a modification to the algorithm that will prevent it from failing.

____

____

(1)

(g) With an ordered array, Items, of many more entries, what more efficient algorithm could be used to achieve your expressed purpose in part (d)?

____

____

(1)

(Total 10 marks)

Q6.

An integer array A contains the following items.

	A
[1]	**3**
[2]	**5**
[3]	**11**
[4]	**12**
[5]	**18**
[6]	**21**
[7]	**26**
[8]	**29**
[9]	**32**

The operator DIV performs integer division. x DIV y calculates how many times y divides exactly into x. For example 7 DIV 3 = 2.

(a) Dry run the following algorithm by completing the trace table.

```
Number ← 12
```

```
Lower ← 1
Upper ← 9
While Lower<Upper
    Current ← (Lower+Upper)DIV 2
    If Number >= A [Current] Then Lower ← Current
    If Number <= A [Current] Then Upper ← Current
EndWhile
Return Current
```

Number	Lower	Upper	Current

Value returned	

(8)

(b) What is the purpose of this algorithm?

____

(1)

(Total 9 marks)

Q7.

A *linear search* and a *binary search* are **two** different methods of searching an ordered list. A given list contains 271 items.

(a) (i) What is the maximum number of items accessed when searching for a particular item from the given list using a linear search?

____

(1)

(ii) Explain your answer.

____

____

(1)

(b) (i) What is the maximum number of items accessed when searching for a particular item from the given list using a binary search?

____

(1)

(ii) Explain your answer.

____

____

(1)

(c) An integer array A contains the following items.

	A
[1]	23
[2]	45
[3]	16
[4]	12
[5]	31

(i) Dry run the following algorithm by completing the trace table.

```
For Count1 ← 1 To 4
   For Count2 ← 1 To 4
      If A[Count2] > A[Count2 + 1] Then
         Temp ← A [ Count2]
         A[Count2] ← A[Count2 + 1]
         A[Count2 + 1] ← Temp
      EndIf
   EndFor
EndFor
```

Count 1	Count2	Temp	A				
			[1]	[2]	[3]	[4]	[5]
–	–	–	23	45	16	12	31
1	1						

(5)

(ii) What is the purpose of this algorithm?

____

(1)

(iii) Suggest **one** way the algorithm could be improved.

____

(1)

(Total 11 marks)

Q8.

The data shown below is a list of surnames of 20 motor car policyholders with the number of claims they have each made in the last five years.

	PolicyHolder		NoOfClaims
1	Wilcox	1	1
2	Adams	2	0
3	Pollard	3	0
4	Williams	4	0
5	Searle	5	3
6	Kelly	6	0

7	Lewis	7	1
8	Franks	8	5
9	Patel	9	1
10	Li Che	10	0
...		...	...
...		...	...
19	Wilkinson	19	3
20	Veale	20	0

(a) (i) The user inputs a policyholder. If the surname is found, the program outputs the number of claims for that policyholder.

```
     Read(SearchName)
     For P := 1 To 20 Do
             If PolicyHolder[P] = SearchName
                Then GoTo 200
     GoTo 300
200 : Write(NoOfClaims[P])
300: End
```

Give **two** reasons why this is badly designed program code.

1. ____

____

2. ____

____ **(2)**

(ii) Write declaration statements (in a language with which you are familiar) for the PolicyHolder or NoOfClaims data structure above, and one other variable used in the code above.

The programming language I am using is ______________________________

1. ____

2. ____ **(2)**

(b) A new task is to design and write code to establish if there are any policyholders who have made five or more claims. The program will output a ‘yes’ or ‘no’ message only.

Write the code for this new task in a programming language with which you are familiar.

(Hint: Use a loop structure to initiate the loop, and then end the loop when some

condition is met.)

(5)

(Total 9 marks)

Q9.

Write down the comparisons needed to look up *Pascal* using a binary search on the following alphabetically sorted list:

Basic, Fortran, Java, Lisp, Pascal, Prolog, Smalltalk.

(Total 3 marks)

Q10.

A recursively-defined procedure X with three integer parameters is defined below.
x DIV *y* calculates how many times *y* divides exactly into *x*. For example 7 DIV 3 = 2.

```
Procedure X (E,L,H)
 If L > H
```

```
                  Then Print 'False'

                  Else M ⟵ (L+H) DIV 2
                    If E = List[M]
                      Then Print 'True'
                      Else
                         If E < List[M]
                           Then X (E,L,M-1)
                           Else X (E,M+1,H)
                         Endif
                    Endif
               Endif
            EndProc
```

(a) What is meant by recursively-defined?

____

(1)

(b) (i) Using the table below, dry-run the procedure call X (6502, 1, 11) applied to the integer array *List* containing the following elements:

Index	**List**
1	1234
2	1789
3	3125
4	4789
5	5006
6	5789
7	6502
8	7411
9	8407
10	8971
11	9053

E	**L**	**H**	**M**	**List[M]**	**Printed Output**

(7)

(ii) What process does procedure X describe?

____

(2)

(Total 10 marks)

Q11.

Write down the comparisons needed to look up *Newcastle* using a binary search on the following list:

Birmingham, Coventry, Liverpool, Manchester, Newcastle, Sheffield, York.

____

____

____

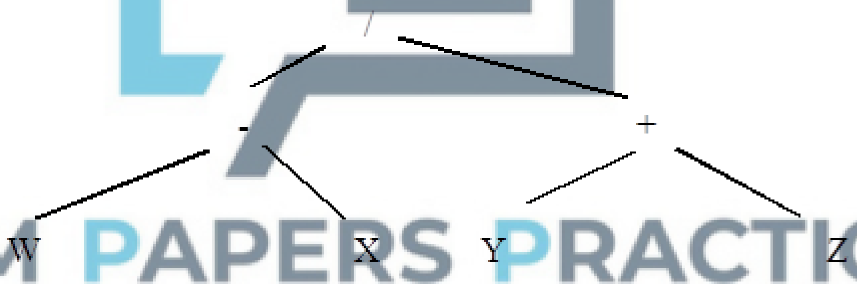
____

____

(Total 3 marks)

Q12.

An algebraic expression is represented in a binary tree as follows.



(a) On the above diagram, circle and label the *root* of this tree, a *branch* and a *leaf node*.

(3)

(b) In the spaces below, draw the *left sub-tree* and the *right sub-tree* of this tree.

left sub-tree	right sub-tree

(2)

(c) What is the result if this tree is printed using in-order traversal?

____

(3)
(Total 8 marks)

Q13.

A *binary* search and a *linear* search are two different methods of searching a list.
A given list contains 137 items.

(a) (i) What is the maximum number of items accessed when searching for a particular item from the given list using a binary search?

____

(ii) Explain your answer.

____

____

(b) (i) What is the maximum number of items accessed when searching for a particular item from the given list using a linear search?

____

(ii) Explain your answer.

____

____

(Total 4 marks)

Q14.

(a) The series of characters J, F, H, U, S, X, T are to be entered into a binary search tree in the order given. Draw a diagram to show how these values will be stored.

(4)

(b) The following data are held in arrays Data, L and R:

Data	'J'	'F'	'H'	'U'	'S'	'X'	'T'
	[1]	[2]	[3]	[4]	[5]	[6]	[7]

L	2	0	0	5	0	0	0
	[1]	[2]	[3]	[4]	[5]	[6]	[7]

R	4	3	0	6	7	0	0
	[1]	[2]	[3]	[4]	[5]	[6]	[7]

Using the arrays above, dry-run the following pseudo-code by completing the trace table opposite:

```
Item ← 'T'

Ptr ← 1
WHILE Data[Ptr] < > Item DO
  PRINT Data[Ptr]
  IF Data[Ptr] > Item

     THEN Ptr ← L[Ptr]

     ELSE Ptr ← R[Ptr]
  ENDIF
ENDWHILE
PRINT Data[Ptr]
```

Trace Table:

Item	Ptr	Printed Output
'T'	1	'J'

(6)
(Total 10 marks)

Q15.

A binary search tree is a data structure where items of data are held such that they can be searched for quickly and easily.
The following data items are to be entered into a binary search tree in the order given:

London, Paris, Rome, Berlin, Amsterdam, Lisbon, Madrid.

(a) Draw a diagram to show how these values will be stored.

(4)

(b) Circle the root node in your diagram.

(1)

(c) If Madrid is being searched for in this binary tree, list the data items which have to be accessed.

____

(1)

(Total 6 marks)

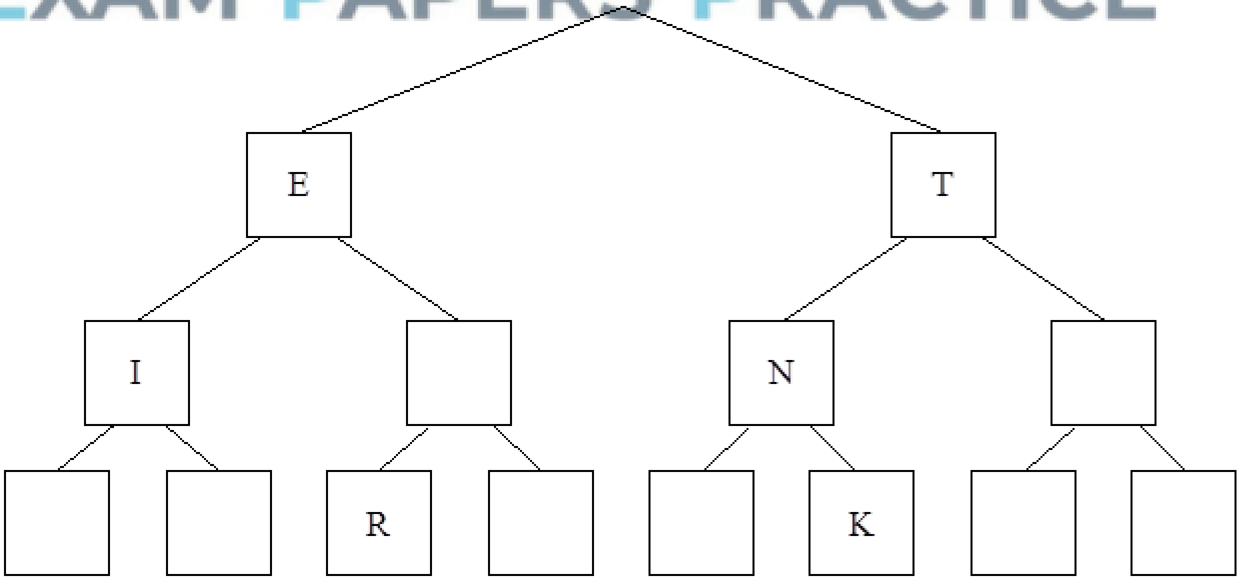
Q16.

A binary tree can be used to represent the alphabet in a code. Part of the tree is shown below. Starting at the root of the tree, branch left is a dot and branch right is a dash.

So N has the code: dash dot.

SOS has the code dot dot dot dash dash dash dot dot dot.

(a) Place the missing letters S and O into the correct positions in the diagram.



(2)

(b) What does the following 2 letter code spell: dot dot dash ?

(2)
(Total 4 marks)

Q17.

The memory of a computer holds an array of records, each of which includes name, address and other information.

(a) What condition is necessary for the binary search (binary chop) process to work correctly?

(2)

(b) Describe this process to find the position in the array of the record containing a given name.

(5)

(c) Why is this search method normally faster than a linear search?

(2)
(Total 9 marks)

Q18.

Using a binary search, a file can be searched more quickly than by linear search.

(i) State the conditions necessary for the binary search to work.

____

____

____

____

(2)

(ii) Briefly describe how this method normally speeds up searching.

____

____

____

____

(2)
(Total 4 marks)

Q19.

The following section of pseudo-code processes a one-dimensional integer array called *List.* The numbers in *List* are stored in ascending order, and x, *Low, High, Middle* are all integer variables. (The function Int returns the whole number part of its parameter.)

```
Proc Process(Low, High, x)
    Found ← False
    Repeat
        Middle ← Int((Low + High)/2)
        If List(Middle) = x
        Then Found ← True
        Else If List(Middle) > x
           Then High ← Middle -1
           Else Low ← Middle +1 {List(Middle) <x}
    Until Found = True
```

(a) Complete the following dry-run table for Process (1, 10, 19), given that the integers in the list are:

2,4, 6, 7, 11, 13, 19, 21, 27, 29

Low	High	Middle	Found
1	10		

(7)

(b) What type of routine does this pseudo-code define?

____

(1)
(Total 8 marks)

EXAM PAPERS PRACTICE