4.4 Classification of algorithms

Name: ______________________

Class: ______________________

Date: ______________________

Time: **130 minutes**

Marks: **91 marks**

Comments:

Q1.

Describe the Halting problem.

____

____

____

____

(Total 2 marks)

Q2.

Why is it not possible to create a Turing machine that solves the Halting problem?

____

____

(Total 1 mark)

Q3.

Employees at a bank use client computers to access data that is stored on a database server.

The database server uses software to query and modify data stored in a database on hard disk drives. It returns the results of these queries to the clients over the bank's computer network.

The performance of the system is unsatisfactory: the time-delay between a client sending a query to the server and the client receiving the results is unacceptably long.

Explain how the performance of the system might be improved. You should consider the following factors that might be affecting the performance:

- the hardware of the server
- the design of the computer network
- the database and software running on the server.

In your answer you will be assessed on your ability to follow a line of reasoning to produce a coherent, relevant and structured response.

(Total 12 marks)

Q4.

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)

Q5.

(a) State the time complexity for the bubble sort algorithm in terms of n, where n is the number of items in the list to be sorted.

___

___

(1)

(b) Explain why the bubble sort algorithm has the time complexity stated in your answer to part **(a)**.

___

___

___

___

(2)

(Total 3 marks)

Q6.

An algorithm is a sequence of unambiguous instructions for solving a problem.

Three different algorithms, A, B and C, have the following orders of time complexity:

Algorithm A: $O(2^n)$
Algorithm B: $O(n)$
Algorithm C: $O(n^3)$

List the algorithms A, B and C in order with the most efficient at the top of the list.

Most efficient: ____________________

____

Least efficient: ____________________

(Total 1 mark)

Q7.

Some problems are tractable.

What does it mean for a problem to be described as tractable?

____

____

(Total 2 marks)

Q8.

One of the problems listed in the table below is tractable.

Place **one** tick next to the tractable problem.

Problem	**Tractable? (Tick one)**
The travelling salesman problem.	
The problem of searching a list.	
The Halting problem.	

(Total 1 mark)

Q9.

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)

Q10.

The internal buses in a computer use parallel communication while most peripherals communicate with a computer using serial communication.

(a) Explain the differences between the ways in which parallel and serial communications carried out.

____

____

____

____

(2)

(b) Most peripherals, such as printers and keyboards, communicate with a computer using a serial connection.

Apart from the widespread availability of USB (Universal Serial Bus) ports, explain why peripherals usually use a serial communication method such as USB instead of parallel communication.

____

____

(1)

(c) In communications systems, a distinction is made between the bit rate and the baud rate.

Define the term baud rate.

____

____

(1)

(d) Explain how it is possible for the bit rate to be higher than the baud rate.

____

____

(1)

(Total 5 marks)

Q11.

Problems can be classified into different categories based upon how efficiently they can be solved, or if they can be solved at all.

Three such classifications are:

- tractable
- intractable
- unsolvable.

(a) Explain what it means for a problem to be described as tractable.

____

____

____

(2)

(b) What approach(es) might a programmer take if asked to ’solve’ an intractable problem?

____

____

____

____

(2)

(c) Tick **one** row in **Table 1** to indicate which of the problems listed in the table is **unsolvable**.

Table 1

Problem	**Unsolvable? (✓ one row)**
The problem of sorting a list into order	
The Halting problem	
The travelling salesman problem	

(1)

(d) Sometimes more than one algorithm exists to solve the same problem.

In such cases, a programmer may select the algorithm to use based upon the time and space complexity of the algorithm.

Table 2 below shows the order of time complexity of three different algorithms to solve a problem.

Tick **one** row in **Table 2** to indicate which of the algorithms is the **least time efficient**.

Table 2

Order of Time Complexity	**Least Time Efficient? (✓ one row)**
$O(2^n)$	
$O(n)$	
$O(n^2)$	

(1)
(Total 6 marks)

Q12.

A computer program stores a list of integers in an array named `List`. The numbers in the array are to be sorted into ascending order so that a particular efficient search algorithm can be used to search for a number.

(a) One of the search algorithms in **Table 1** can only be used successfully on a sorted list.

Place **one** tick next to the name of the algorithm that requires a list to be sorted.

Table 1

Algorithm Name	Requires Sorted List? (Tick one box)
Binary search	
Linear search	

(1)

(b) The pseudo-code for a standard algorithm that can be used to sort the data in the array `List` into order is shown in **Figure 1**. The variable `ListLength` stores a count of the number of items in the array `List`.

Array indexing starts at 1.

Figure 1

```
For OuterPointer ← 2 To ListLength
  CurrentValue ← List[OuterPointer]
  InnerPointer ← OuterPointer - 1
  While InnerPointer > 0 And
        List[InnerPointer] > CurrentValue Do
    List[InnerPointer + 1] ← List[InnerPointer]
    InnerPointer ← InnerPointer - 1
  EndWhile
  List[InnerPointer + 1] ← CurrentValue
EndFor
```

Complete the empty (unshaded) cells in the trace table (**Table 2**) for an execution of the algorithm in **Figure 1** when the array `List` contains the values `9, 8, 5` and `6` in that order.

Table 2

				List			
List Length	**Outer Pointer**	**Current Value**	**Inner Pointer**	[1]	[2]	[3]	[4]
				9	8	5	6
4	2		1				
			0				
	3		2				
			1				
			0				
	4		3				

			2				
			1				

(3)

(c) In the trace table (**Table 2**), when the variable `OuterPointer` contains the value `2` and then `3`, the value of the variable `InnerPointer` decreases to `0`. When `OuterPointer` contains `4`, `InnerPointer` stops decreasing when it reaches the number `1`.

Explain why `InnerPointer` does not decrease to `0` when `OuterPointer` contains `4`.

____

____

____

____

(1)

(d) Tick **one** box in **Table 3** to indicate the correct Order of **Time** Complexity of the standard algorithm in **Figure 1**.

Table 3

Order of Time Complexity	**Tick one box**
O(n)	
$O(n^2)$	
$O(2^n)$	

(1)

(e) State the name of the standard algorithm that is represented by the pseudo-code in **Figure 1**.

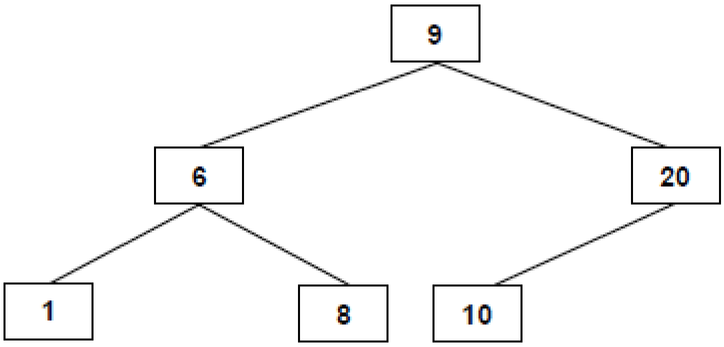
____

(1)

(f) Instead of storing a list of numbers in an array as in (b), the numbers could be stored in a binary search tree. This would also enable efficient searching.

The numbers 9, 6, 1, 8, 20 and 10 are put into a binary search tree in that order. **Figure 2** shows this binary search tree.

Figure 2



(i) A search of the binary tree is performed for the number 8.

List the numbers, in the order that they would be checked, for the search to determine that the number 8 **is present** in the tree.

(1)

(ii) A search of the binary tree is performed for the number 11.

List the numbers, in the order that they would be checked, for the search to determine that the number 11 **is not present** in the tree.

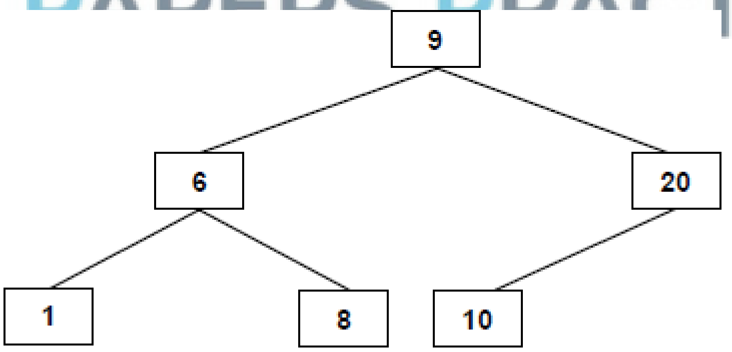
(1)

(g) The numbers 4, 5 and 3 are to be added into the binary search tree, in that order.

Figure 3 below is an identical copy of **Figure 2**.

Complete **Figure 3** below to show the binary search tree from **Figure 2** after the extra numbers have been added into it.

Figure 3



(2)

(Total 11 marks)

Q13.

An algorithm is a sequence of unambiguous instructions for solving a problem.

(a) Three different algorithms, A, B and C, have the following orders of time complexity:

Algorithm A: $O(a^n)$

Algorithm B: $O(n^2)$
Algorithm C: $O(n)$

List the algorithms A, B and C in order with the most efficient at the top of the list.

Most efficient: _______________

____

Least efficient: _______________

(1)

(b) Some problems are intractable.

(i) What does it mean for a problem to be described as *intractable*?

____

____

____

(2)

(ii) One of the problems listed in the table below is intractable.

Place **one** tick next to the intractable problem.

Problem	**Intractable? (Tick one)**
The travelling salesman problem	
The problem of sorting a list of names into alphabetic order	
The Halting problem	

(1)
(Total 4 marks)

Q14.

(a) Time complexity is one of the two measures that are used to describe the complexity of an algorithm.

What is the other measure?

____

(1)

(b) A student has been asked to write a program to list duplicate entries in a file containing a list of words. The diagram below shows her first attempt at planning an algorithm. The algorithm will not work in all circumstances.

```
Open file
N ← Number of items in file
For Pos1 ← 1 To N Do
```

```
    Read item at position Pos1 in file into variable W1
    For Pos2 ← 1 To N Do
      Read item at position Pos2 in file into variable W2
      If W1 = W2 And Not (Pos1 = Pos2)
        Then Output 'Duplicate: ' , W1
      EndIf
    EndFor
  EndFor
  Close file
```

The basic operation in the algorithm is the `If` statement that compares two words. The contents of a particular file are shown in the table below.

File position	Item
1	Rope
2	Dagger
3	Rope

(i) Complete the table below by tracing the execution of the algorithm in the diagram above when it is applied to the file in the table above.

N	Pos1	W1	Pos2	W2	Output

(3)

(ii) Tick **one** box in the table below to indicate the correct order of time complexity of the algorithm that the student has written.

Order of time complexity	Tick one box
$O(a^n)$	
$O(n)$	
$O(n^2)$	

(1)

(iii) Justify your answer to part (ii).

____

____

____

(2)
(Total 7 marks)

Q15.

(a) Complete the missing parts of the question posed by the Halting problem in the diagram below.

Is it possible in general to ____ that

can tell, given any program and its inputs and without

______________________________________, whether the given program with

its given inputs will halt?

(2)

(b) What is the significance of the Halting problem?

____

____

(1)
(Total 3 marks)

Q16.

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.

(1)

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

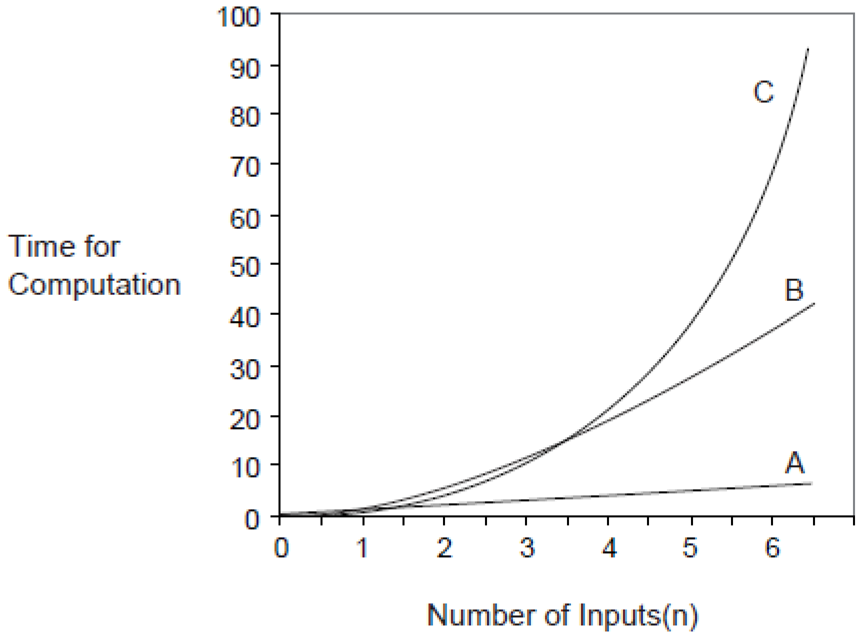
Order of time complexity	Tick one box
$O(\log_2 n)$	
$O(n)$	
$O(n^2)$	

(1)

(Total 5 marks)

Q17.

The graph below illustrates the time complexity of three different algorithms, A, B and C.



(a) The three algorithms have orders of time complexity $O(n^2)$, $O(n)$ and $O(a^n)$.

(i) What is the order of time complexity of algorithm C? ____________

(1)

(ii) Which of the algorithms, A, B or C, is the most time efficient? ______

(1)

(b) The Travelling Salesman problem is intractable.

(i) What is meant by an *intractable* problem?

____

____

____

(2)

(ii) What approach might a programmer take if asked to 'solve' an intractable problem?

____

____

____

(2)

(Total 6 marks)

Q18.

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)