0 1 . 1 There are four athletes taking part in a race. Each athlete is allocated a lane that they run in. There can only be one athlete in each lane and there are four lanes available.

How many different permutations are there for allocating athletes to lanes?

[1 mark]

0 1 . 2 If there are *n* athletes taking part in a race and *n* lanes available, with each athlete being allocated to one lane and each lane only used by one athlete, how many different permutations are there for allocating athletes to lanes?

[1 mark]

0 1 . 3 An anagram of a string is another string that contains exactly the same number of each character as in the original string but the characters may not all be in the same positions as in the original string. The original string is also an anagram of itself.

If there are *n* characters in a string, why will the number of anagrams of the string not always be the same as your answer to question **03.2**?

[1 mark]

0 1 . 4 The travelling salesperson problem (TSP) is when a salesperson has to visit every city in a set of cities and needs to find the shortest route that does this without visiting any city more than once before finally returning to the city that they started from.

The TSP is an example of an intractable problem.

Explain what is meant by an intractable problem.

[2 marks]

Figure 3 lists some time complexities, where *n* is the size of the problem input and *k* denotes a constant.

Figure 3

$O(1)$
$O(n^k)$
$O(k^n)$
$O(n)$
$O(\log n)$
$O(n \log n)$

0 1 . 5 Assuming that the time complexities in **Figure 3** are for the most time efficient algorithm that solves a problem, how many of the time complexities in **Figure 3** are for intractable problems?

[1 mark]

0 1 . 6 State which of the time complexities shown in **Figure 3** is the time complexity of the **linear** search algorithm and explain why it has that time complexity.

[2 marks]

0 1 . 7 State which of the time complexities shown in **Figure 3** is the time complexity of the **binary** search algorithm and explain why it has that time complexity.

[2 marks]

0 2 Big-O notation is used to express the time complexity of an algorithm. **Table 1** contains a list of algorithms.

State the Big-O time complexity of each of these algorithms. The first row has been completed for you.

Table 1

Algorithm	**Time complexity**
Binary tree search	O(log n)
Bubble sort	
Linear search	
Merge sort	

Copy the contents of the unshaded cells in **Table 1** into the table in your Electronic Answer Document.

[3 marks]

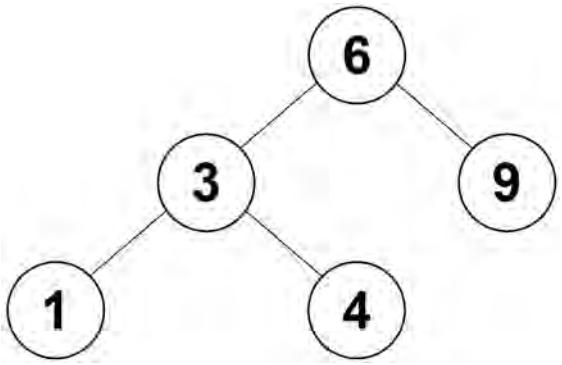
0 3 A binary tree is a type of data structure.

0 3 . 1 State **two** characteristics that make a tree a binary tree.

[2 marks]

0 3 . 2 **Figure 2** shows a binary tree and its representation using an array of records called `Tree`. Each record consists of three fields, `Data`, `Left` and `Right`.

Figure 2



	Data	Left	Right
[0]	6	1	4
[1]	3	2	3
[2]	1	-1	-1
[3]	4	-1	-1
[4]	9	-1	-1

Figure 3 shows a subroutine that implements a binary tree search algorithm using the array `Tree`. The subroutine parameter, `k`, is the data item being searched for. The subroutine returns a Boolean value indicating if the data item being searched for is in the binary tree or not.

Parts of the algorithm are missing.

Figure 3

```
SUBROUTINE BTS(k)
    Current ← _____1_____
    WHILE Current > _____2_____
        IF Tree[Current].Data = k THEN
            RETURN _____3_____
        ELSEIF Tree[Current].Data < k THEN
            _____4_____
        ELSE
            _____5_____
        ENDIF
    ENDWHILE
    RETURN _____6_____
ENDSUBROUTINE
```

Complete each row in **Table 1**, to show what the labels indicating missing parts of the algorithm in **Figure 3** should be replaced by.

Table 1

1	
2	
3	
4	
5	
6	

Copy the contents of the unshaded cells in **Table 1** into the table in your Electronic Answer Document.

[4 marks]

0 3 . 3 There are similarities in how the binary tree search and the binary search algorithms work.

State the big-O time complexity of the **binary search** algorithm.

[1 mark]

0 3 . 4 Explain why the binary search algorithm has the time complexity stated in your answer for **03.3**.

[1 mark]

0 3 . 5 Explain why searching for an item in a list or tree is a tractable problem.

[1 mark]

0 3 . 6 Heuristics can be used when working with an intractable problem.

Explain what heuristics are.

[2 marks]

0 3 . 7 Explain what is meant by constant time complexity.

[2 marks]