0 1 **Figure 1** shows a binary tree containing seven nodes. **Figure 2** shows how the binary tree in **Figure 1** could be represented using three one-dimensional arrays: `Data`, `Dir1` and `Dir2`.

Figure 1

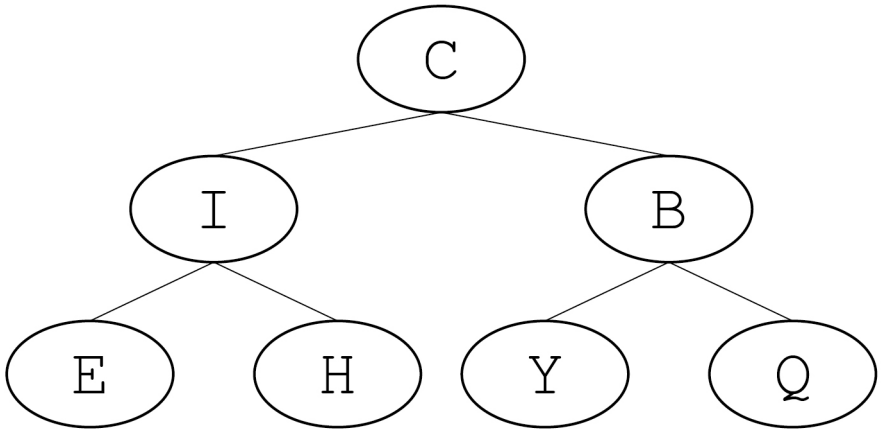


Figure 2

Index	Data	Dir1	Dir2
[0]	C	1	4
[1]	I	2	3
[2]	E	-1	-1
[3]	H	-1	-1
[4]	B	5	6
[5]	Y	-1	-1
[6]	Q	-1	-1

0 1 . 1 Define the term binary tree.

[2 marks]

0 1 . 2 The output of a post-order traversal algorithm used to print the data item at each node in the binary tree shown in **Figure 1** would be `E`, `H`, `I`, `Y`, `Q`, `B`, `C`.

State the output that would be produced by an **in-order** traversal algorithm.

[2 marks]

Figure 3 shows pseudo-code for a subroutine called `Traversal` that uses the three arrays from **Figure 2**.

Figure 3

```
SUBROUTINE Traversal(StartNode)
  Current ← StartNode
  Pos ← 0
  Stack[Pos] ← Current
  WHILE Pos ≠ -1
    Current ← Stack[Pos]
    Pos ← Pos - 1
    OUTPUT Data[Current]
    IF Dir2[Current] ≠ -1 THEN
      Pos ← Pos + 1
      Stack[Pos] ← Dir2[Current]
    ENDIF
    IF Dir1[Current] ≠ -1 THEN
      Pos ← Pos + 1
      Stack[Pos] ← Dir1[Current]
    ENDIF
  ENDWHILE
ENDSUBROUTINE
```

0 1 . 3 Complete the unshaded cells in **Table 2** to show the result of the subroutine call `Traversal(0)`

Table 2

		Stack				
Current	**Pos**	**[0]**	**[1]**	**[2]**	**[3]**	**Output**

Copy the contents of the unshaded cells in **Table 2** into the table in your Electronic Answer Document.

[7 marks]

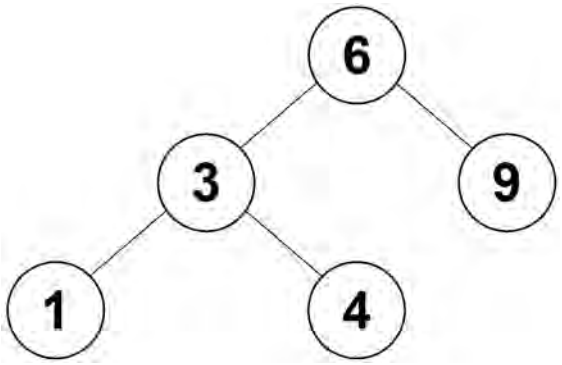
0 2 A binary tree is a type of data structure.

0 2 . 1 State **two** characteristics that make a tree a binary tree.

[2 marks]

0 2 . 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 2 . 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 2 . 4 Explain why the binary search algorithm has the time complexity stated in your answer for **03.3**.

[1 mark]

0 2 . 5 Explain why searching for an item in a list or tree is a tractable problem.

[1 mark]

0 2 . 6 Heuristics can be used when working with an intractable problem.

Explain what heuristics are.

[2 marks]

0 2 . 7 Explain what is meant by constant time complexity.

[2 marks]

0 3 **Figure 3** shows a graph containing five nodes. **Figure 4** shows how the graph in **Figure 3** could be represented using three one-dimensional arrays: `Data`, `Dir1` and `Dir2`.

Figure 3

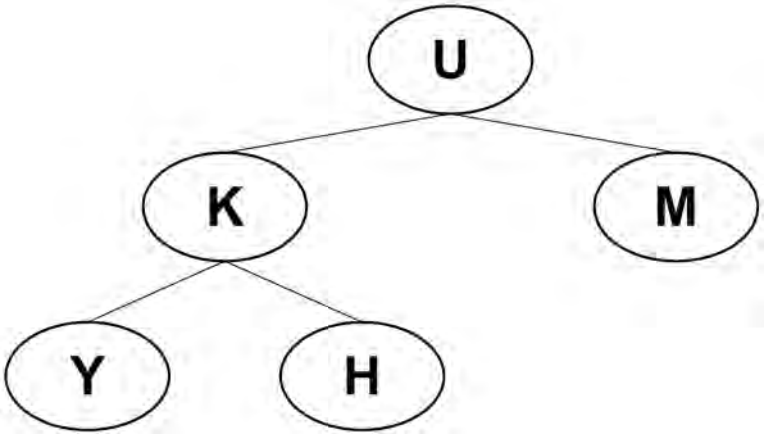


Figure 4

Index	Data	Dir1	Dir2
[0]	U	1	4
[1]	K	2	3
[2]	Y	−1	−1
[3]	H	−1	−1
[4]	M	−1	−1

0 3 . 1 The graph in **Figure 3** is a binary tree. A binary tree is a rooted tree where each node has at most two child nodes.

There are three properties that a graph needs to have for it to be a tree. One of those properties is that it contains no cycles.

State the other **two** properties of this graph that make it a tree.

[2 marks]