AQA Computer Science A-Level

4.3.5 Sorting algorithms

Past Paper Questions

Additional Spec Qs Paper 1

2 **Figure 1** contains pseudo-code for a recursive merge sort algorithm. **Figure 2** contains pseudo-code for an algorithm called `Merge` that is called by the merge sort algorithm in **Figure 1**.

Figure 1

```
FUNCTION MergeSort(L, S, E)
  IF S < E THEN
    M ← (S + E) DIV 2
    L1 ← MergeSort(L, S, M)
    L2 ← MergeSort(L, M + 1, E)
    RETURN Merge(L1, L2)
  ELSE
    RETURN Append([], L[S])
  ENDIF
ENDFUNCTION
```

Figure 2

```
FUNCTION Merge(L1, L2)
  L3 ← []
  WHILE Len(L1) > 0 AND LEN(L2) > 0
    IF L1[1] < L2[1] THEN
      L3 ← Append(L2[1], L3)
      L2 ← RemoveFirstItem(L2)
    ELSE
      L3 ← Append(L1[1], L3)
      L1 ← RemoveFirstItem(L1)
    ENDIF
  ENDWHILE
  WHILE Len(L1) > 0
    L3 ← Append(L1[1], L3)
    L1 ← RemoveFirstItem(L1)
  ENDWHILE
  WHILE Len(L2) > 0
    L3 ← Append(L2[1], L3)
    L2 ← RemoveFirstItem(L2)
  ENDWHILE
  RETURN L3
ENDFUNCTION
```

The `RemoveFirstItem` function takes a list and returns a list that contains all the items in the original list except the first one. For example, if `Names` is the list `["Gemma", "Richard", "Georgina", "Margaret"]` then the function call `RemoveFirstItem(Names)` will return the list `["Richard", "Georgina", "Margaret"]`.

The `Len` function takes a list and returns the number of items that are in the list. For example, if `Names` is the list `["Gemma", "Richard", "Georgina", "Margaret"]` then the function call `Len(Names)` will return the value of `4`.

The `Append` function takes an item and a list and returns a list that has all the items from the original list followed by the item. For example, if `Names` is the list `["Gemma", "Richard", "Georgina", "Margaret"]` then the function call `Append("Matt", Names)` will return the list `["Gemma", "Richard", "Georgina", "Margaret", "Matt"]`.

The first item in the list has an index of 1.

0 2 . 3 Complete **Table 1** to show the result of tracing the `MergeSort` algorithm shown in **Figure 1** with the function call `MergeSort(ListToSort, 1, 5)`. `ListToSort` is the list `[6, 3, 4, 8, 5]`. The first six rows and the **Call number** column have been completed for you.

Copy your answer in **Table 1** into the Electronic Answer Document.

Table 1

Call number	S	E	M	List returned
1	1	5	3	
2	1	3	2	
3	1	2	1	
4	1	1		[6]
3	1	2	1	
5	2	2		[3]
3				
2				
6				
2				
1				
7				
8				
7				
9				
7				
1				

[6 marks]

0 2 . 4 What is the time complexity for the `MergeSort` algorithm shown in **Figure 1**?

[1 mark]

A stack frame is used with subroutine calls.

0 2 . 5 State **two** items that will be stored in a stack frame for a subroutine call.

[2 marks]

When the subroutine call `MergeSort(ListToSort, 1, 5)` is made four is the largest number of stack frames, generated by this subroutine call, that will be on the stack at any one time.

0 2 . 6 Explain why there will be **three** occasions when there will be four stack frames on the stack when the subroutine call `MergeSort(ListToSort, 1, 5)` is made.

[2 marks]

June 2011 Comp 1

The contents of an array Scores are shown in **Figure 2**.

A pseudo code representation of an algorithm is given in **Figure 3**.

Figure 2

Scores							
[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
18	23	36	21	58	40	45	59

Figure 3

```
Max ← 8
FOR Count1 ← 1 TO (Max - 1) DO
  FOR Count2 ← 1 TO (Max - 1) DO
    IF Scores[Count2] > Scores[Count2 + 1]
      THEN
        Temp ← Scores[Count2]
        Scores[Count2] ← Scores[Count2 + 1]
        Scores[Count2 + 1] ← Temp
    ENDIF
  ENDFOR
ENDFOR
```

1 8 **One pass** is made through the outer loop of the algorithm in **Figure 3**.

Complete **Table 2** to show the changed contents of the array `Scores` after this single pass. You may use **Table 3** to help you work out your answer, though you are neither required to use **Table 3** nor to copy it into your Electronic Answer Document.

Copy the bottom row of your completed ***Table 2*** *into the Electronic Answer Document.*

Table 2

Scores							
[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]

Table 3

Max	Count1	Count2	Temp	Scores							
				[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
				18	23	36	21	58	40	45	59

(4 marks)

1 9 What is the name of the standard algorithm shown in **Figure 3**? *(1 mark)*