MODULE L STANDARD ALGORITHMS · L4

Merge sort

Original practice questions · 20 marks · ~25 minutes · OCR J277 spec 2.1.3

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **20 marks**.

1

Total: 3 marks

(a) Tick (✓) **one** box in each row to show whether each statement about a merge sort is true or false. **[3]**

Statement	True	False
A list that contains only one item is already sorted.		
A merge sort can only be used if the data is already in order.		
When two lists are merged, they are joined end to end and the new list is then sorted.		

2

Total: 2 marks

(a) A merge sort has two main stages. Describe the **two** main stages of a merge sort. **[2]**

1

2

3 **Total: 4 marks**

A parcel depot weighs eight parcels before loading a van. The heaviest parcels must be loaded first. The weights, in kg, are stored in this order:

9	27	3	14	21	5	11	18

(a) Show the steps that a merge sort would take to put these weights into **descending** order (largest first). **[4]**

4 **Total: 4 marks**

A wildlife trust stores the names of animals it has recorded. Part way through a merge sort, these two lists are both in alphabetical order and are about to be merged into one list in alphabetical order.

Left list	Badger	Heron	Otter	Wren
Right list	Crane	Fox	Stoat	Vole

(a) Complete the table to show the next **three** comparisons made during the merge, and the item that is moved into the new list after each one. The first comparison has been done for you. **[3]**

Comparison	Items compared	Item moved into the new list
1	*Badger and Crane*	*Badger*
2		
3		
4		

(b) Give the new list when the merge is complete. **[1]**

5 **Total: 5 marks**

A programmer at the parcel depot writes this function. It is given two lists, `left` and `right`, which are each already in ascending order. Arrays are indexed from 0.

```
01 function combine(left, right)
02     array result[left.length + right.length]
03     i = 0
04     j = 0
05     k = 0
06     while i < left.length AND j < right.length
07         if left[i] <= right[j] then
08             result[k] = left[i]
09             i = i + 1
10         else
11             result[k] = right[j]
12             j = j + 1
13         endif
14         k = k + 1
15     endwhile
16     while i < left.length
17         result[k] = left[i]
18         i = i + 1
19         k = k + 1
20     endwhile
21     while j < right.length
22         result[k] = right[j]
23         j = j + 1
24         k = k + 1
25     endwhile
26     return result
27 endfunction
```

(a) Identify the sorting algorithm that this function is part of. **[1]**

(b) Describe the purpose of lines 16 to 25. **[2]**

(c) The function is called with `left = [4, 9, 15]` and `right = [2, 10]`.

(i) State the contents of `result` when the function returns. **[1]**

(ii) State how many times line 07 is run during this call. **[1]**

Answer: .....

6 **Total: 2 marks**

At the end of each day the depot sorts a list of about 50 000 parcel codes. It could use a bubble sort or a merge sort.

(a) Give **one** advantage of using a merge sort instead of a bubble sort for this list. **[1]**

(b) Give **one** disadvantage of using a merge sort instead of a bubble sort. **[1]**

Total for this worksheet: 20 marks