2.3 ALGORITHMS · 2.3.1(F)

Merge & quick sort

Original practice questions · 30 marks · about 40 minutes · spec 2.3.1(f)

Instructions. Answer all questions. Marks are shown in brackets [ ]. Show splits/partitions clearly.

1

Total: 6 marks

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

(b) State the time complexity of a merge sort. _____ **[1]**

(c) Explain why a merge sort is described as a divide-and-conquer algorithm. **[2]**

(d) State one disadvantage of a merge sort. _____ **[1]**

2

Total: 6 marks

```
[8, 3, 5, 1]
```

(a) Show how a merge sort would **split** this list down to single items, then **merge** them back. Draw the split/merge clearly. **[4]**

(b) State the final sorted list. _____ **[1]**

(c) State the result of merging [3, 8] and [1, 5]. _____ **[1]**

3 Total: 6 marks

```
[6, 2, 8, 1, 4] (use the last item, 4, as the pivot)
```

(a) After partitioning around the pivot 4, state the items to its left and to its right. **[2]**

(b) Explain what is special about the pivot's position after partitioning. **[2]**

(c) State what quick sort does next, after the first partition. **[2]**

4 Total: 6 marks

(a) State the average time complexity of quick sort. **[1]**

(b) State quick sort's worst-case complexity, and when it occurs. **[2]**

(c) State **two** differences between merge sort and quick sort. **[2]**

(d) State which of the two can sort "in place". **[1]**

5 Total: 6 marks

(a) Explain why merge sort and quick sort are far better than bubble sort for a list of 1,000,000 items. **[3]**

(b) A system has very little spare memory. State which of merge sort and quick sort is more suitable and why. **[2]**

(c) State the number of "levels" of splitting for a merge sort on a list of 8 items. **[1]**

END OF QUESTIONS