

2.3 ALGORITHMS · 2.3.1(F)

Bubble & insertion sort

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

Instructions. Answer all questions. Marks are shown in brackets []. Show the list after each pass / insertion.**1** Total: 6 marks

(a) Describe how a bubble sort sorts a list. [2]

(b) State the worst-case time complexity of a bubble sort. [1]

(c) State how a bubble sort can detect that the list is already sorted and stop early. [2]

(d) State whether a bubble sort needs a second list (is it "in place"?). [1]

2 Total: 7 marks

[6, 3, 7, 1, 4]

(a) Show the list **after each complete pass** of a **bubble sort**, until it is sorted. [5]

after pass 1					
after pass 2					
after pass 3					
after pass 4					

(b) State which value is guaranteed to be in its correct place after pass 1. [1]

(c) State the largest number of comparisons in one pass of a 5-item list. [1]

3

Total: 6 marks

[6, 3, 7, 1, 4]

- (a) Show the list **after each insertion** of an **insertion sort**. [4]

insert 2nd item					
insert 3rd item					
insert 4th item					
insert 5th item					

- (b) Describe, in words, how an insertion sort places each new item. [2]

4

Total: 5 marks

- (a) State one similarity between bubble sort and insertion sort. [1]

- (b) State one difference between how bubble sort and insertion sort work. [2]

- (c) State the best-case complexity of these sorts on an already-sorted list, and why. [2]

5

Total: 6 marks

- (a) Explain why bubble sort and insertion sort are both $O(n^2)$. [2]

- (b) A list of 1,000,000 items must be sorted quickly. Explain why a bubble sort would be a poor choice, and name a better algorithm. [2]

- (c) Give one situation where bubble sort or insertion sort would still be a reasonable choice. [2]

END OF QUESTIONS