4.1 Decision Maths: Algorithms, Sorting and Bin Packing: Test Pack

Total Marks

EDEXCEL 9FM0 · Calculator allowed · about 60 minutes

Name: _____ Date: ____ / ____ / _____

Answer ALL questions. Show all your working.

1 The following list of seven numbers is to be sorted into ascending order:
58, 23, 91, 12, 67, 45, 34

(a) Perform a bubble sort on the list to sort it into ascending order. Show the list in full after each pass, and continue until you have completed one full pass in which no swaps are made. **(4)**

(b) State the total number of passes needed and the total number of comparisons made across all of these passes. **(2)**

(c) State the total number of swaps made during the whole sort, and explain why a bubble sort must include a final pass with no swaps before you can be certain the list is fully sorted. **(2)**

(Total for Question 1 is 8 marks)

© Revision Library | revisionlibrary.co.uk

2 The following pseudocode describes an algorithm applied to a list of n numbers:

Step 1: Let List = [a list of n numbers]
Step 2: Let Max = List(1), Min = List(1)
Step 3: For i = 2 to n
Step 4: If List(i) > Max then Max = List(i)
Step 5: If List(i) < Min then Min = List(i)
Step 6: Next i
Step 7: Let Range = Max - Min
Step 8: Output Max, Min, Range

(a) State, in your own words, what this algorithm calculates for a given list of n numbers. **(1)**

(b) Trace this algorithm for List = [23, 45, 12, 67, 34, 9, 58], recording the values of Max and Min after each value of i from 2 to 7 is considered. Present your working as a trace table. **(3)**

(c) State the values output by the algorithm for Max, Min and Range. **(1)**

(d) State the order of this algorithm in terms of n, the number of items in the list, and justify your answer. **(2)**

(Total for Question 2 is 7 marks)

3 This question compares how a bubble sort performs on the same set of six numbers arranged in two different orders.

(a) The marks (out of 60) scored by six students in a Further Maths mock exam are recorded in the order the papers were marked: 36, 14, 58, 22, 41, 9. Sort this list into ascending order using a bubble sort, showing the list in full after each pass. Continue until a pass with no swaps occurs, and state the total number of passes and the total number of swaps used. **(5)**

(b) The same six marks are instead given in the order 9, 14, 22, 36, 58, 41. Sort this list into ascending order using a bubble sort, showing the list after each pass, and state the number of passes needed this time. **(3)**

(c) By comparing your answers to parts (a) and (b), explain what this shows about the best-case and worst-case performance of a bubble sort on a list of n items. **(2)**

(Total for Question 3 is 10 marks)

© Revision Library | revisionlibrary.co.uk

4 The items with the following sizes are to be packed into bins of capacity 40: 21, 5, 11, 17, 36, 30, 15, 28.

(a) Find a lower bound for the number of bins needed. **(2)**

(b) Use the first-fit algorithm, taking the items in the order given, to pack the items into bins. State the contents of each bin. **(3)**

(c) Use the first-fit decreasing algorithm to pack the items. State the number of bins used and the contents of each bin. **(3)**

(Total for Question 4 is 8 marks)

© Revision Library | revisionlibrary.co.uk

5 The following list shows the scores (out of 100) achieved by seven contestants in a coding competition:
72, 55, 89, 41, 67, 60, 78

(a) Using median-of-three pivot selection (at each stage, compare the first, middle and last items of the current sublist, and choose whichever of these three values is the median as the pivot), perform a quicksort on this list to sort it into ascending order. At each stage, state the three items compared, the pivot chosen, and the two resulting sublists. Write out the final sorted list. **(5)**

(b) Give one reason why choosing the pivot by the median-of-three method, rather than always using the first item of the sublist, can improve the efficiency of a quicksort, and describe a type of list for which this improvement is most noticeable. **(2)**

(Total for Question 5 is 7 marks)

---

6 Tomasz is helping his family move house. Some of the boxes have the following weights, in kg, and must be packed into crates that can each hold a maximum of 9 kg:
6, 5, 6, 5, 6, 6, 5, 5

(a) Calculate the theoretical lower bound for the number of boxes needed to pack these items. **(2)**

(b) Sort the list of weights into decreasing order and apply first-fit decreasing, packing into crates of capacity 9 kg. Show the contents of each crate, and state the number of crates used. **(3)**

(c) Explain why no packing method, including first-fit decreasing, can achieve the lower bound found in part (a) for this particular set of weights. **(2)**

(Total for Question 6 is 7 marks)

© Revision Library | revisionlibrary.co.uk

Mark scheme · 4.1 Decision Maths: Algorithms, Sorting and Bin Packing: Test Pack

Question 1

(a) correct method for pass 1: at least 4 of the 6 adjacent comparisons and resulting swaps correctly applied, e.g. list becomes 23, 58, 12, 67, 45, 34, 91 oe **M1**

(a) passes 2 and 3 both fully correct: 23, 12, 58, 45, 34, 67, 91 then 12, 23, 45, 34, 58, 67, 91 **A1**

(a) pass 4 fully correct, giving the sorted list 12, 23, 34, 45, 58, 67, 91 cao **A1**

(a) a further (5th) pass shown in which no swaps occur, confirming the list is fully sorted **B1**

(a) Answer: Pass 1: 23, 58, 12, 67, 45, 34, 91. Pass 2: 23, 12, 58, 45, 34, 67, 91. Pass 3: 12, 23, 45, 34, 58, 67, 91. Pass 4: 12, 23, 34, 45, 58, 67, 91. Pass 5 (check): 12, 23, 34, 45, 58, 67, 91 (no swaps).

(b) 5 passes cao **B1**

(b) 30 comparisons cao (ft from their number of passes: 6 comparisons per pass) **B1**

(b) Answer: 5 passes; 30 comparisons.

(c) 12 swaps cao (5 + 3 + 3 + 1 + 0) **B1**

(c) a pass with no swaps means every adjacent pair in the list is already in the correct order, so no further pass can change the list, confirming it is sorted (oe) **B1**

(c) Answer: 12 swaps in total. A pass with no swaps confirms every adjacent pair is correctly ordered, so the list cannot change further and must be sorted.

Question 2

(a) it finds the maximum value, the minimum value, and the range (Max - Min) of the list (oe) **B1**

(a) Answer: It finds the maximum value, the minimum value, and the range of the list of numbers.

(b) correct initial values before the loop: Max = 23, Min = 23 **M1**

(b) correct values through i = 2 to i = 4: i=2 Max=45 Min=23; i=3 Max=45 Min=12; i=4 Max=67 Min=12 **A1**

(b) correct values through i = 5 to i = 7: i=5 Max=67 Min=12 (no change); i=6 Max=67 Min=9; i=7 Max=67 Min=9 (no change) **A1**

(b) Answer: i: 2,3,4,5,6,7 giving Max: 45,45,67,67,67,67 and Min: 23,12,12,12,9,9.

(c) Max = 67, Min = 9, Range = 58, all correct **B1**

(c) Answer: Max = 67, Min = 9, Range = 58.

(d) order O(n), i.e. linear **B1**

(d) the loop runs a fixed number of comparisons (2 comparisons, at Steps 4 and 5) for each of the n - 1 remaining items, so the total number of comparisons is directly proportional to n (oe) **B1**

(d) Answer: O(n): the loop performs a fixed number of comparisons for each of the n - 1 remaining items, so the total work is directly proportional to n.

Question 3

(a) pass 1 correct: 14, 36, 22, 41, 9, 58 oe **M1**

(a) passes 2 to 5 all correct, reaching the sorted list 9, 14, 22, 36, 41, 58 **A1**

(a) a further (6th) pass shown with no swaps, confirming the list is sorted **B1**

(a) 6 passes stated cao **B1**

(a) 9 swaps stated cao (ft from their trace) **B1**

(a) Answer: Sorted list: 9, 14, 22, 36, 41, 58. 6 passes; 9 swaps.

(b) pass 1 correct: identifies that only 58 and 41 are out of order, giving 9, 14, 22, 36, 41, 58 **M1**

(b) pass 2 shown with no swaps, confirming the list is sorted **A1**

(b) 2 passes stated cao **B1**

(b) Answer: Sorted list (unchanged apart from the last pair): 9, 14, 22, 36, 41, 58, found in 2 passes.

(c) the best case occurs when the list is already sorted, or nearly sorted (as in part (b)), needing very few passes (as few as 2, i.e. close to O(n) comparisons overall) (oe) **B1**

(c) the worst case occurs when the list is far from sorted (as in part (a)), needing up to n - 1 passes and $O(n^2)$ comparisons in total (oe) **B1**

(c) Answer: Best case: a nearly-sorted list needs only a small, fixed number of passes. Worst case: a list far from sorted can need up to n - 1 passes, giving $O(n^2)$ total comparisons.

Question 4

(a) total size = 163 **M1**

(a) 163/40 = 4.08, so at least 5 bins **A1**

(a) Answer: 5

(b) places each item in the first bin with enough room, in the order given **M1**

(b) first bins correct **A1**

(b) bins: (21, 5, 11); (17, 15); (36); (30); (28) **A1**

(b) Answer: (21, 5, 11); (17, 15); (36); (30); (28)

(c) sorts the items into descending order **M1**

(c) applies first-fit to the sorted list **M1**

(c) 5 bins: (36); (30, 5); (28, 11); (21, 17); (15) **A1**

(c) Answer: 5 bins: (36); (30, 5); (28, 11); (21, 17); (15)

© Revision Library | revisionlibrary.co.uk

Question 5

(a) correctly identifies the first, middle and last items of the full list as 72, 41 and 78, and selects the median value, 72, as the first pivot **B1**

(a) correct first split: {55, 41, 67, 60} (< 72) and {89, 78} (≥ 72) **A1**

(a) correct second pivot on {55, 41, 67, 60}: first/middle/last are 55, 41, 60, median is 55, giving splits {41} and {67, 60} **A1**

(a) correctly orders the remaining 2-item sublists: {67, 60} becomes 60, 67 and {89, 78} becomes 78, 89 **A1**

(a) correct final sorted list 41, 55, 60, 67, 72, 78, 89 cao **A1**

(a) Answer: Final sorted list: 41, 55, 60, 67, 72, 78, 89.

(b) median-of-three is far less likely to choose an extreme (smallest or largest) value as the pivot, giving more balanced splits on average (oe) **B1**

(b) the improvement is most noticeable on a list that is already sorted, or nearly sorted, in ascending or descending order, where the first-item method would otherwise produce the worst-case $O(n^2)$ behaviour (oe) **B1**

(b) Answer: Median-of-three avoids consistently unbalanced splits, keeping the algorithm closer to its efficient O(n log n) average-case behaviour, most noticeably on lists that are already sorted or nearly sorted.

Question 6

(a) total weight = 6+5+6+5+6+6+5+5 = 44 kg **M1**

(a) lower bound = ceiling(44/9) = 5 cao **A1**

(a) Answer: Lower bound = 5 crates.

(b) correct decreasing order: 6, 6, 6, 6, 5, 5, 5, 5 **M1**

(b) correctly shows that no two items can share a crate (6+6=12 > 9, 6+5=11 > 9, 5+5=10 > 9), so each item is placed in its own crate **A1**

(b) 8 crates used, cao **B1**

(b) Answer: Every item needs its own crate: 8 crates used.

(c) any two of these items sum to more than the crate capacity of 9 kg (6+6=12, 6+5=11, 5+5=10, all exceed 9), so no crate can ever hold more than one item (oe) **B1**

(c) this means the true minimum number of crates is 8 (one per item), so the lower-bound formula (total weight / capacity, rounded up) only gives a necessary condition, not a value that is always achievable (oe) **B1**

(c) Answer: Since every pair of items sums to more than 9 kg, no crate can ever hold two items, so 8 crates are always needed regardless of method; the lower bound of 5 is not achievable for this data.

© Revision Library | revisionlibrary.co.uk