MODULE L STANDARD ALGORITHMS · L3

Bubble sort: mark scheme

21 marks · OCR J277 spec 2.1.3 (searching and sorting algorithms: bubble sort) and 2.1.2 (trace tables)

How to use. 1 mark per bullet up to the Max shown. **/** separates alternative answers. Words in (brackets) are not needed for the mark. **...** means the mark depends on the bullet before it. J277 does not require students to remember bubble sort code: credit understanding of the steps and of the given code. Code answers may be in OCR Exam Reference Language or any high-level language; ignore minor syntax slips.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) J277 2.1.3	Compares items next to each other: True Only used if data already partly in order: False Largest value in final position after first pass: True Splits the list into sub-lists of one item: False	**4**	1 mark per correct row. Both boxes ticked in a row: 0 for that row. Row 2 tests the pre-requisite: a bubble sort has none (it works on unsorted data). Row 4 describes merge sort.
2(a) J277 2.1.3	After pass 1: 7, 18, 12, 9, 25, 30 After pass 2: 7, 12, 9, 18, 25, 30 After pass 3: 7, 9, 12, 18, 25, 30	**3**	1 mark per correct row. Follow through: credit pass 2 or pass 3 if it is one correct bubble sort pass of the student's own previous row. Do not credit a row that is simply the fully sorted list unless it follows from the previous row (e.g. no marks for writing 7, 9, 12, 18, 25, 30 in every row).
2(b) J277 2.1.3	The algorithm only stops when a (whole) pass makes no swaps Pass 3 made a swap (9 and 12) / swaps were made in pass 3 ... ... so it cannot tell / does not know the list is sorted until it checks again Pass 4 makes no swaps, so the sort then stops / ends	**2**	Max 2. Do not accept "to check it is sorted" alone for 2 marks (1 mark only if no other point is made). Do not accept "it always does one pass per item" or "it does a fixed number of passes": the stem states it stops when a pass makes no swaps.

<table>
<tr><th>QUESTION</th><th>ANSWER</th><th>MARKS</th><th>GUIDANCE</th></tr>
<tr><td>3(a)
J277 2.1.3</td><td>Compares adjacent items / each item with the next one / <code>prices[i]</code> with <code>prices[i + 1]</code> (line 04)
Swaps the two items if they are in the wrong order (lines 05 to 07)
Goes through the whole list in a pass / inner (for) loop through the list (line 03)
Repeats passes until a pass makes no swaps / uses a flag (<code>swapped</code>) to stop (lines 01, 02, 08, 11)</td><td>2</td><td>Max 2. Line numbers are not needed. Do not accept "it sorts the prices" or "it puts them in order" (outcome, not a feature). Do not accept "it uses a loop" alone.</td></tr>
<tr><td>3(b)
J277 2.1.3</td><td>Records / shows whether a swap has been made (during a pass)
Set to false at the start of each pass and to true when a swap happens
Used to decide when to stop / the loop repeats until a pass makes no swaps / it stays false when the list is sorted</td><td>2</td><td>Max 2. Do not accept "to swap the items" (that is <code>temp</code>'s job) or "it counts the swaps".</td></tr>
<tr><td>3(c)
J277 2.1.3 / 2.1.2</td><td><code>if prices[i] < prices[i + 1] then</code></td><td>1</td><td>Accept <code>if prices[i + 1] > prices[i] then</code>. Accept the comparison written in any high-level language. Allow a missing <code>then</code>. Do not accept <= (equal prices would be swapped every pass, so the loop would never end if two prices are the same). Do not accept "change > to <" with no line written: the question asks for the line.</td></tr>
<tr><td>4(a)
J277 2.1.2 / 2.1.3</td><td>1 mark: <code>i</code> takes 0, 1, 2 (in order)
1 mark: <code>temp</code> takes 14 then 21
1 mark: <code>swaps</code> takes 0, 1, 2 (in order)
1 mark: Output 2 (once, at the end)
<table><tr><th>i</th><th>temp</th><th>swaps</th><th>Output</th></tr><tr><td></td><td></td><td>0</td><td></td></tr><tr><td>0</td><td>14</td><td>1</td><td></td></tr><tr><td>1</td><td></td><td></td><td></td></tr><tr><td>2</td><td>21</td><td>2</td><td></td></tr><tr><td></td><td></td><td></td><td>2</td></tr></table></td><td>4</td><td>Mark each column as a sequence of values; rows may be laid out differently. Values may be repeated on later rows (e.g. swaps 1 repeated when i = 1). Do not credit <code>i</code> = 3. Do not credit <code>temp</code> = 9 or 6. Output must appear once only; "x" or "-" in the Output column counts as an output.</td></tr>
<tr><td>4(b)
J277 2.1.3</td><td>9, 14, 6, 21</td><td>1</td><td>All four in this order. Accept with or without brackets / commas. Do not accept the fully sorted list 6, 9, 14, 21 (only one pass is made).</td></tr>
</table>

QUESTION	ANSWER	MARKS	GUIDANCE
4(c) J277 2.1.3	Line 06 overwrites `speeds[i]` / the first item is overwritten (by the second) ... so its original value would be lost / both positions would end up holding the same value `temp` keeps a copy of the first item ... ... so it can be put into `speeds[i + 1]` / the second position (line 07)	**2**	Max 2. Credit an example using the data, e.g. "without temp, 14 would be lost and both would be 9". Do not accept "to store a temporary value" alone, or "to make the swap faster".

Total: 21 marks