MODULE L STANDARD ALGORITHMS · L2

Binary search: mark scheme

21 marks · OCR J277 spec 2.1.3 (searching algorithms: main steps, segments of code, pre-requisites, apply to a data set, identify from code) and 2.1.2 (trace tables)

How to use. 1 mark per bullet up to the Max shown. **/** separates alternative words; **//** separates alternative answers for the same mark. Words in (brackets) are not needed for the mark. **...** at the start of a bullet means it depends on the bullet before it. Candidates are not required to remember the code for a binary search (J277 2.1.3).

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) J277 2.1.3	4, 9, 15, 22, 31, 48: Yes "Ash", "Elm", "Fir", "Oak", "Pine", "Yew": Yes 12, 5, 18, 3, 27, 30: No	**3**	1 mark per correct row. Two ticks in a row: 0 for that row. Row 2 is in alphabetical order, so it is sorted. Row 3 is not sorted.
1(b) J277 2.1.3	The list / data must be sorted / in order	**1**	Accept: in ascending / descending / alphabetical / numerical order. Do not accept "the list must be organised" or "the item must be in the list".
2(a) J277 2.1.3	167 is the (first) middle item checked / compared with 205 205 is greater than 167, so 167 and the items to its left / 103 to 167 are discarded // search continues with 171, 186, 194, 205, 219 194 is the next middle item; 205 is greater, so 171, 186 (and) 194 are discarded // search continues with 205, 219 205 is the next middle item ... it matches / equals the target, so it is found / the search stops	**4**	Max 4. Accept the stages shown as a series of lists with the middle item marked. Do not award a discard mark if the wrong half is discarded (FT not allowed for the half kept). Allow the item to the right of centre if used consistently: 167, then 194, then 219 (205 < 219, so 219 discarded), then 205 found. Award the same four points in this order. Do not credit a description of a linear search (checking each item in turn).

<table>
<tr><th>QUESTION</th><th>ANSWER</th><th>MARKS</th><th>GUIDANCE</th></tr>
<tr><td>2(b)(i)
J277 2.1.3</td><td>167, 125 (in this order)
... 140 (last, and nothing else)</td><td>2</td><td>Correct answer: 167, 125, 140. 2 marks for all three in this order and no others; 1 mark for 167, 125 in order. Allow 167, 125, 152, 140 (taking the item right of centre in a part-list of 2) for 2 marks.</td></tr>
<tr><td>2(b)(ii)
J277 2.1.3</td><td>There are no items left to check // the part of the list left to search is empty // the lower pointer / position passes the upper pointer / position</td><td>1</td><td>Accept: after 140 is discarded nothing is left between 125 and 140. Do not accept "it reaches the end of the list" (that is a linear search) or "130 is not found" alone.</td></tr>
<tr><td>3(a)
J277 2.1.3</td><td>Binary search
... it finds the middle position / item (line 05) // it compares the middle item with the target (lines 06 / 08) // it moves <code>low</code> or <code>high</code> to discard half of the array (lines 09 / 11) // it repeats while <code>low <= high</code></td><td>2</td><td>Reason mark only if binary search is identified. A line number alone is not enough; the reason must say what the code does. Do not accept "the array is sorted" (that is a pre-requisite, not something in the code) or "it searches the array".</td></tr>
<tr><td>3(b)
J277 2.1.3
J277 2.1.2</td><td><table><tr><th>LOW</th><th>HIGH</th><th>MID</th><th>VALUE RETURNED</th></tr><tr><td>0</td><td>7</td><td>3</td><td></td></tr><tr><td></td><td>2</td><td>1</td><td></td></tr><tr><td>2</td><td></td><td>2</td><td>2</td></tr></table><code>low</code>: 0, 2
<code>high</code>: 7, 2
<code>mid</code>: 3, 1, 2
Value returned: 2 (only)</td><td>4</td><td>1 mark per column: each variable's values in the correct order. Ignore the row a value is written on and ignore repeated values on later rows. Value returned: 2 (the index), not 23. Do not award the last mark if -1 is also returned.</td></tr>
<tr><td>3(c)
J277 2.1.3</td><td>It discards the lower half / the middle item and everything before it (because the target is larger than the middle item) // the next search starts just after the middle position</td><td>1</td><td>Accept "moves the start of the search to after mid". Do not accept "adds 1 to low" alone (repeats the code without saying why).</td></tr>
<tr><td>3(d)
J277 2.1.3</td><td>The target / serial number is not in the array / was not found</td><td>1</td><td>Accept: "-1 is not a valid index, so it shows the number is not stored". Do not accept "an error has happened".</td></tr>
</table>

QUESTION	ANSWER	MARKS	GUIDANCE
4(a) J277 2.1.3	Binary search halves the number of records left / discards half of the remaining records with each comparison Linear search checks the records one at a time / in turn (from the start) ... so a linear search may have to check all / most of the 100 000 records Binary search needs at most about 17 comparisons for 100 000 records	**2**	Max 2. Do not accept "binary search is faster / quicker" alone (it repeats the question). Do not accept "binary search starts in the middle" alone without the halving. Credit any other valid point. The number 17 is not needed; accept "about 16 or 17".

Total: 21 marks