This section presents you various set of Mock Tests related to **Data Structures Algorithms**. You can download these sample mock tests at your local machine and solve offline at your convenience. Every mock test is supplied with a mock test key to let you verify the final score and grade yourself.

### DATA STRUCTURES ALGORITHMS MOCK TEST IV

<table>
<thead>
<tr>
<th>Question</th>
<th>Options</th>
</tr>
</thead>
</table>
| **Q 1 - Recursion uses more memory space than iteration because** | A - it uses stack instead of queue.  
B - every recursive call has to be stored.  
C - both A & B are true.  
D - None of the above are true. |
| **Q 2 - Heap is an example of** | A - complete binary tree  
B - spanning tree  
C - sparse tree  
D - binary search tree |
| **Q 3 - In a min heap** | A - minimum values are stored.  
B - child nodes have less value than parent nodes.  
C - parent nodes have less value than child nodes.  
D - maximum value is contained by the root node. |
| **Q 4 - In the deletion operation of max heap, the root is replaced by** | A - next available value in the left sub-tree. |
**Q 5 - All possible spanning trees of graph G**

A - have same number of edges and vertices.
B - have same number of edges and but not vertices.
C - have same number of vertices but not edges.
D - depends upon algorithm being used.

**Q 6 - From a complete graph, by removing maximum ____________ edges, we can construct a spanning tree.**

A - e-n+1  
B - n-e+1  
C - n+e-1  
D - e-n-1

**Q 7 - If we choose Prim's Algorithm for uniquely weighted spanning tree instead of Kruskal's Algorithm, then**

A - we'll get a different spanning tree.  
B - we'll get the same spanning tree.  
C - spanning will have less edges.  
D - spanning will not cover all vertices.

**Q 8 - Re-balancing of AVL tree costs**

A - O1  
B - Ologn  
C - O\(n\)  
D - O\(n^2\)

**Q 9 - A balance factor in AVL tree is used to check**

A - what rotation to make.  
B - if all child nodes are at same level.  
C - when the last rotation occurred.  
D - if the tree is unbalanced.
Q 10 - Binary search tree is an example of complete binary tree with special attributes.
A - BST does not care about complete binary tree properties.
B - BST takes care of complete binary tree properties.
C - It depends upon the input.
D - None of the above.

Q 11 - The following sorting algorithms maintain two sub-lists, one sorted and one to be sorted –
A - Selection Sort
B - Insertion Sort
C - Merge Sort
D - both A & B

Q 12 - If locality is a concern, you can use ______ to traverse the graph.
A - Breadth First Search
B - Depth First Search
C - Either BFS or DFS
D - None of the above!

Q 13 - Access time of a binary search tree may go worse in terms of time complexity upto
A - O(n²)
B - O(nlog n)
C - O(n)
D - O(1)

Q 14 - Shell sort uses
A - insertion sort
B - merge sort
C - selection sort
D - quick sort

Q 15 - A pivot element to partition unsorted list is used in
A - Merge Sort
B - Quick Sort
C - Insertion Sort
D - Selection Sort

Q 16 - A stable sorting algorithm –
A - does not crash.
B - does not run out of memory.
C - does not change the sequence of appearance of elements.
D - does not exist.

Q 17 - An adaptive sorting algorithm –
A - adapts to new computers.
B - takes advantage of already sorted elements.
C - takes input which is already sorted.
D - none of the above.

Q 18 - Interpolation search is an improved variant of binary search. It is necessary for this search algorithm to work that –
A - data collection should be in sorted form and equally distributed.
B - data collection should be in sorted form and but not equally distributed.
C - data collection should be equally distributed but not sorted.
D - None of the above.

Q 19 - If the data collection is in sorted form and equally distributed then the run time complexity of interpolation search is –
A - O(n)
B - O(1)
C - O(log n)
D - O(log(log n))

Q 20 - Which of the following algorithm does not divide the list –
A - linear search
B - binary search
C - merge sort
D - quick sort

Q 21 - The worst case complexity of binary search matches with –
Q 22 - Apriori analysis of an algorithm assumes that –
A - the algorithm has been tested before in real environment.
B - all other factors like CPU speed are constant and have no effect on implementation.
C - the algorithm needs not to be practical.
D - none of the above.

Q 23 - Aposterior analysis are more accurate than apriori analysis because –
A - it contains the real data.
B - it assumes all other factors to be dynamic.
C - it assumes all other factors to be constant.
D - it is a result of reverse-engineering.

Q 24 - Project scheduling is an example of
A - greedy programming
B - dynamic programming
C - divide and conquer
D - none of the above.

Q 25 - In conversion from prefix to postfix using stack data-structure, if operators and operands are pushed and popped exactly once, then the run-time complexity is –
A - O1
B - O\text{n}
C - O\log n
D - O(n^2)

ANSWER SHEET

<table>
<thead>
<tr>
<th>Question Number</th>
<th>Answer Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
</tr>
<tr>
<td>4</td>
<td>D</td>
</tr>
<tr>
<td>5</td>
<td>A</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>B</td>
</tr>
<tr>
<td>8</td>
<td>B</td>
</tr>
<tr>
<td>9</td>
<td>D</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
</tr>
<tr>
<td>11</td>
<td>D</td>
</tr>
<tr>
<td>12</td>
<td>B</td>
</tr>
<tr>
<td>13</td>
<td>C</td>
</tr>
<tr>
<td>14</td>
<td>A</td>
</tr>
<tr>
<td>15</td>
<td>B</td>
</tr>
<tr>
<td>16</td>
<td>C</td>
</tr>
<tr>
<td>17</td>
<td>B</td>
</tr>
<tr>
<td>18</td>
<td>A</td>
</tr>
<tr>
<td>19</td>
<td>D</td>
</tr>
<tr>
<td>20</td>
<td>A</td>
</tr>
<tr>
<td>21</td>
<td>B</td>
</tr>
<tr>
<td>22</td>
<td>B</td>
</tr>
<tr>
<td>23</td>
<td>A</td>
</tr>
<tr>
<td>24</td>
<td>B</td>
</tr>
<tr>
<td>25</td>
<td>B</td>
</tr>
</tbody>
</table>