**Answer to Theory**

1. The deque module is part of the Python Standard Library, which is basically located within the collections module. It allows both deletion and insertion from both ends of the queue.
2. A tree is a set of nodes and edges while a graph is a set of vertices and edges.

A tree has root which is a unique node while a graph has no root.

1. The time complexity of an algorithm measures the duration it takes for the algorithm to execute based on the size of the input. Likewise, the space complexity of an algorithm measures the amount of memory or space it requires to run based on the size of the input. The time complexity is measured in terms of steps processed while space complexity is measured in terms of storage units that was acquired.
2. Bubble Sort is an uncomplicated algorithm that relies on comparisons to rearrange elements in a list. It iteratively scans through the list and exchanges adjacent elements if they are not in the correct order. This procedure is repeated until the list is completely sorted.

The element that belongs at the end of the sorted list, also known as the largest element, is ensured to be in its appropriate position at the conclusion of the list.

1. LIFO and FIFO are two commonly utilized data structures in programming. LIFO, also known as 'last in, first out,' refers to a data structure where the most recent element added to the stack is processed first. Conversely, FIFO, or 'first in, first out,' refers to a data structure where the initial element added to the queue is processed first.
2. A binary tree is considered balanced if it satisfies three conditions:

* Firstly, the height of the left and right subtrees for any node should not differ by more than 1.
* Secondly, the left subtree of that node should also be balanced.
* Lastly, the right subtree of that node should also be balanced.

How to search in a binary tree is as written below:

* **Begin:** Initiate the exploration from the root node.
* **Compare:** Evaluate the search value in relation to the value at the current node. If the search value matches the value of the current node, the search is deemed successful. If the search value is lesser than the value of the current node, proceed to the left subtree. If the search value is greater than the value of the current node, proceed to the right subtree.
* **Continue:** Persist in comparing and traversing left or right subtrees based on the comparisons until the search value is discovered or until a leaf node (end of the tree) is reached. Search
* **Result:** If the search value is found, return the node that contains the value. If the value is not found in the tree, the search concludes, indicating the absence of the value.