Ans 1. An array stores a fixed-size sequential collection of elements of the same type, whereas list is a generic collection.

Ans 2. A binary tree consists of a number of nodes that contain the data to be stored (or pointers to the data), and the following structural characteristics : Each node has up to two direct child nodes. There is exactly one node, called the root of the tree, that has no parent node. All other nodes have exactly one parent.

Ans3.

1. Perform inorder traversal of tree1 and store each node's value in arr1.
2. Perform inorder traversal of tree2 and store each node's value in arr2.
3. Combine arr1 and arr2 using merge function of merge sort to create result array.
4. Return result array.

Ans4.

A Heap is a special Tree-based data structure in which the tree is a complete binary tree. Generally, Heaps can be of two types: Max-Heap: In a Max-Heap the key present at the root node must be greatest among the keys present at all of it's children

Ans5.

In computing, a hash table (hash map) is a data structure that implements an associative array abstract data type, a structure that can map keys to values. A hash table uses a hash function to compute an index, also called a hash code, into an array of buckets or slots, from which the desired value can be found.

Ans6.

1. Time complexity is a function describing the amount of time an algorithm takes in terms of the amount of input to the algorithm. ...
2. Space complexity is a function describing the amount of memory (space) an algorithm takes in terms of the amount of input to the algorithm.

Ans7.

Sort Stack using Recursion

A stack is a linear data structure that operates on the Last In First Out (LIFO) principle. This indicates that the last thing added to the stack is deleted first. The alternate word for a stack is LIFO, which refers to the order in which items are removed from it (last in, first out). This form of structure is called a "stack" because it resembles a stack of real goods placed on top of each other. This structure makes it simple to remove one item from the top of the stack, but removing an item from the bottom of the stack may need to remove numerous other things first. A peek action may also provide access to the top of the stack without altering it.

Sorting a stack is useful for various tasks, including memory management, maintaining the context of a process in the event of an interrupt, and other high-priority tasks. Although we shall see the recursive approach here, sorting may also be done iteratively.

Operations in Stack

In a stack, we can perform the following operations, such as:

* Push(): Pushes an element into the stack
* Pop(): Pops out the top element of the stack
* Top(): Returns the top element of the stack
* Size(): Returns the size of the stack

A stack is a very useful and crucial data structure that is employed in memory management and process flow scheduling. The program counter is one of the most important applications of the stack because it saves the context of a processor code from stacking if it has to transition to a new process so that it may return to the old process and finish it when the new process is completed.

*Recursion* is one of the most significant algorithms because if we can solve a smaller work, we can almost certainly solve the entire project utilizing the smaller jobs. Recursion is a term that refers to the act of calling oneself. It has a base case, which is the major case, in which it handles the smaller problem scenario before calling itself for the minor sections. It uses the fact that while we are standing in a certain state, we presume that our recursive function has done processing for smaller responses, which we can now combine to solve our present state.