**DATA STRUCTURES**

1. **What is a Data Structure?**

The data structure is a way that specifies how to organize and manipulate the data. It also defines the relationship between them. Some examples of Data Structures are arrays, Linked List, Stack, Queue, etc. Data Structures are the central part of many computer science algorithms as they enable the programmers to handle the data in an efficient way

1. **What are linear and non-linear data Structures?**

**Linear:** A data structure is said to be linear if its elements form a sequence or a linear list. In linear data structures, the elements are stored in a non-hierarchical way where each item has the successors and predecessors except the first and last element.

Examples: Array. Linked List, Stacks and Queues

**Non-Linear:**The Non-linear data structure does not form a sequence i.e. each item or element is connected with two or more other items in a non-linear arrangement. The data elements are not arranged in the sequential structure.

Example: Graph and Trees.

1. **What are the various operations that can be performed on different Data Structures?**

* **Insertion** Add a new data item in the given collection of data items.
* **Deletion** Delete an existing data item from the given collection of data items.
* **Traversal** Access each data item exactly once so that it can be processed.
* **Searching** Find out the location of the data item if it exists in the given collection of data items.
* **Sorting** Arranging the data items in some order i.e. in ascending or descending order in case of numerical data and in dictionary order in case of alphanumeric data.

1. **How array is differ from linked list?**

* The size of the arrays is fixed, Linked Lists are Dynamic in size.
* Inserting and deleting a new element in an array of elements is expensive, Whereas both insertion and deletion can easily be done in Linked Lists.
* Random access is not allowed in Linked Listed.
* Extra memory space for a pointer is required with each element of the Linked list.
* Arrays have better cache locality that can make a pretty big difference in performance

1. **What is Stack and where it can be used?**

Stack is a linear data structure which the order LIFO(Last In First Out) or FILO(First In Last Out) for accessing elements. Basic operations of the stack are: **Push, Pop, Peek**

1. **What is a Queue, how it is different from the** **stack and how is it implemented?**

Queue is a linear structure that follows the order is **F**irst **I**n **F**irst **O**ut (FIFO) to access elements. Mainly the following are basic operations on queue: **Enqueue , Dequeue**, **Front, Rear**  
 The difference between stacks and queues is in removing. In a stack we remove the item the most recently added; in a queue, we remove the item the least recently added. Both Queues and Stacks can be implemented using Arrays and Linked Lists.

1. **What is a Linked List and What are its types?**

A linked list is a linear data structure (like arrays) where each element is a separate object. Each element (that is node) of a list is comprising of two items – the data and a reference to the next node. Types of Linked List :

1. **Singly Linked List :**In this type of linked list, every node stores address or reference of next node in list and the last node has next address or reference as NULL. For example 1->2->3->4->NULL
2. **Doubly Linked List :**Here,here are two references associated with each node, One of the reference points to the next node and one to the previous node. Eg. NULL<-1<->2<->3->NULL.
3. **Circular Linked List :**Circular linked list is a linked list where all nodes are connected to form a circle. There is no NULL at the end. A circular linked list can be a singly circular linked list or doubly circular linked list. Eg. 1->2->3->1 [The next pointer of last node is pointing to the first]
4. **What is a Stack?**

Stack is an ordered list in which, insertion and deletion can be performed only at one end that is called the top. It is a recursive data structure having pointer to its top element. The stack is sometimes called as Last-In-First-Out (LIFO) list i.e. the element which is inserted first in the stack will be deleted last from the stack.

1. **What are the operations that can be performed on a stack?**

* Push Operations
* Pop Operations
* Peek Operations

**10.What is the difference between PUSH and POP?**

PUSH and POP operations specify how data is stored and retrieved in a stack.

**PUSH:** PUSH specifies that data is being "inserted" into the stack.

**POP:** POP specifies data retrieval. It means that data is being deleted from the stack.

**11.What is an array?**

Arrays are defined as the collection of similar types of data items stored at contiguous memory locations. It is the simplest data structure in which each data element can be randomly accessed by using its index number.

**12. Define the tree data structure.**

The Tree is a recursive data structure containing the set of one or more data nodes where one node is designated as the root of the tree while the remaining nodes are called as the children of the root. The nodes other than the root node are partitioned into the nonempty sets where each one of them is to be called sub-tree.

**13. What are Binary trees?**

### In each node can have atmost two children. Binary tree is generally partitioned into three disjoint subsets, i.e. the root of the node, left sub-tree and Right binary sub-tree.

**14.  Define the graph data structure?**

A graph G can be defined as an ordered set G(V, E) where V(G) represents the set of vertices and E(G) represents the set of edges which are used to connect these vertices. A graph can be seen as a cyclic tree, where the vertices (Nodes) maintain any complex relationship among them instead of having parent-child relations.

**15. What is a Binary Search Tree?**

* A binary search tree (BST) is a variant of binary tree data structure that stores data in a very efficient manner such that the values of the nodes in the left sub-tree are less than the value of the root node, and the values of the nodes on the right of the root node are correspondingly higher than the root.
* Also, individually the left and right sub-trees are their own binary search trees at all instances of time.

**16. What is an AVL Tree?**

AVL trees are **height (self) balancing** Binary Search Tree. AVL tree checks the height of left and right sub-trees and assures that the difference is **not more than 1**. This difference is called Balance Factor and is calculates as.

BalanceFactor = height(left subtree) − height(right subtree)

**17. What is the difference between tree and graph data structure?**

Tree and graph are differentiated by the fact that a tree structure must be connected and can never have loops whereas in the graph there are no restrictions.

Tree provides insights on relationship between nodes in a hierarchical manner and graph follows a network model.