**Assignment 5**

**2. Mention the types of Data structure?**

Ans: A data structure is a collection of data type ‘values’ which are stored and organized in such a way that it allows for efficient access and modification. In some cases a data structure can become the underlying implementation for a particular data type.

For example, composite data types are data structures that are composed of scalar data types and/or other composite types, whereas an abstract data type will define a set of behaviours (almost like an ‘interface’ in a sense) for which a particular data structure can be used as the concrete implementation for that data type.

When we think of data structures, there are generally four forms:

1. **Linear**: arrays, lists
2. **Tree**: binary, heaps, space partitioning etc.
3. **Hash**: distributed hash table, hash tree etc.
4. **Graphs**: decision, directed, acyclic etc.

**3. Explain nonlinear data structure?**

Ans: Non linear data structure is a data structure in which data items are not stored linearly in the memory. So there is no contiguous memory allocation of the data. This feature is included because it uses the memory optimally.

memory is represented in blocks. So each time a certain amount of memory is allocated for storing any data. If that data is smaller than the allocated memory, the extra memory is wasted as that memory is not big enough to store another data. But may be it can store a part of data. So this feature helps in the way that the memory that got wasted like this can be used to store a part of the data, and the remaining parts are stored in some other memory location and all allocated memory is connected, like this the memory is saved from being wasted and our data also remains intact in order. So this non linear data structure decreases the space complexity and the memory is used optimally.

**1.     Linked list**

Linked list is a non linear data structure in which data is stored in memory with contiguous memory allocation. One item of linked list is linked with next data item. Linked list are of different types- circular linked list, doubly linked list, circular doubly linked list.

The one advantage of linked list it allows insertion, deletion at any position of the linked list. A circular doubly linked list is the one in which the last data item is linked with the first data item. Whereas a doubly linked list is the one in which there are both forward and backward links between data items.

**2.     Trees**

A tree is  an abstract data type, having a hierarichal structure. A tree has root node, and a sub tree. It follows parent – child relationship.

Types of trees are-

* Binary tree:- A binary tree is a tree in which each node has at most two child nodes, one left child and one right child. It is used to implement binary search tree.
* AVL tree:- Adlen veliskii landis (avl) tree is a self  balanced tree in which  the difference between the height of the two child sub trees at any node is 0, 1, -1. The tree is balanced to maintain its property in case of any disperancy.
* Binary search tree
* Complete binary tree
* Red – black tree
* Weighted balance tree
* B tree – A B tree can have more than two children at a node.
* B+ tree
* 2 -3 tree

**3.     Graph**

A graph is an abstract data structure which consists finite set of vertices and edges. A graph may be directed or undirected.

**Advantages**

* Easy to understand.
* Contiguous memory allocation is not required.

**Disadvantages**

* Overhead of link is requir

**4. What are the differences between stack and queue?**

Ans: Difference between Stack and Queue

|  |  |
| --- | --- |
| Stack | Queue |
| The stack is based on LIFO(Last In First Out) principle | The queue is based on FIFO(First In First Out) principle. |
| Insertion Operation is called Push Operation | Insertion Operation is called Enqueue Operation |
| Deletion Operation is called Pop Operation | Deletion Operation is called Dequeue Operation |
| Push and Pop Operation takes place from one end of the stack | Enqueue and Dequeue Operation takes place from a different end of the queue |
| The most accessible element is called Top and the least accessible is called the Bottom of the stack | The insertion end is called Rear End and the deletion end is called the Front End. |
| Simple Implementation | Complex implementation in comparison to stack |
| Only one pointer is used for performing operations | Two pointers are used to perform operations |
| Empty condition is checked using  Top==-1 | Empty condition is checked using  Front==-1||Front==Rear+1 |
| Full condition is checked using  Top==Max-1 | Full condition is checked using  Rear==Max-1 |
| There are no variants available for stack | There are three types of variants i.e circular queue, double-ended queue and priority queue |
| Can be considered as a vertical collection visual | Can be considered as a horizontal collection  visual |
| Used to solve the recursive type problems | Used to solve the problem having sequential processing |

**5. What are the application of queue?**

Ans: A queue data structure is generally used in scenarios where the FIFO approach (First In First Out) has to be implemented. The following are some of the most common applications of queue in data structure:

* Managing requests on a single shared resource such as CPU scheduling and disk scheduling
* Handling hardware or real-time systems interrupts
* Handling website traffic
* Routers and switches in networking
* Maintaining the playlist in media players