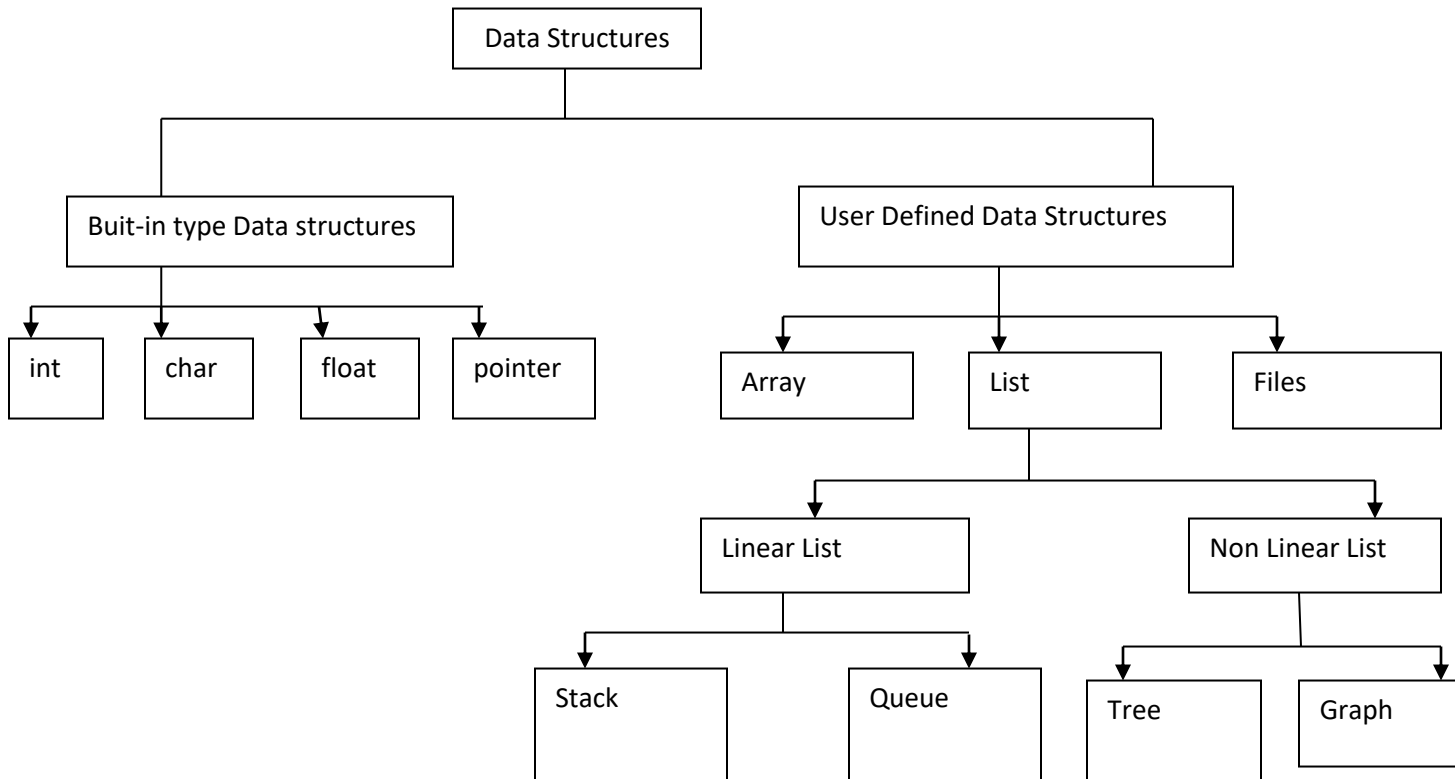


Classification of Data Structure



1. Built-in type Data Structure(Primitive Data Structure): Primitive data structure is a data structure which is directly operated upon by the machine instructions.

Following are the Primitive data structures:

- i. **Int:** is a primitive data structure which accepted integer type value. Example `int x=20`.
- ii. **Char:** is a primitive data structure which accepted character type value. `Char x='a'`.
- iii. **Float:** is a primitive data structure which accepted float type value.
- iv. **Pointer:** pointer holds the address of the next variable in memory.

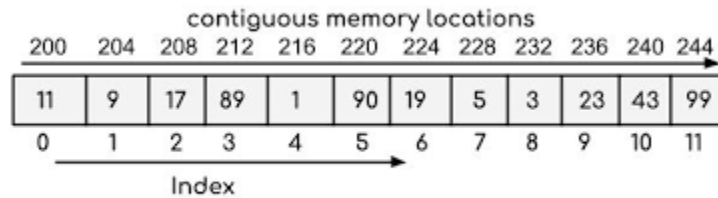
2. User Defined Data Structures(Non Primitive Data Structures): Non Primitive data structures are not directly operated upon by the machine instructions.

Non primitive data structures are derived from the primitive data structures.

Following are the non primitive data structures:

- i. **Array:** Array is a collection of homogeneous (same type) data item.

Example `int A[12]`



- ii. **Files:** File is a collection of various records of the single entity. File is used to organize data in secondary storage device such as hard disk.

- iii. **List:** List stores data elements in sequence. List is of two types:

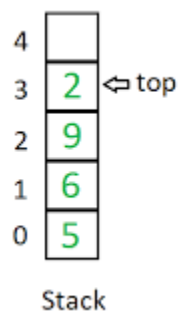
- a. **Linear List:** Linear list or Linear data structure forms a sequence means all its elements are arranged in a linear order.

Linear Data structures are of three types:

- i. **Stack:** Stack is a linear data structure in which insertion and deletion can take place only at one end called top.

Stack works on the principle of Last in First out (LIFO).

Example: Deck of cards, Piles of plates.



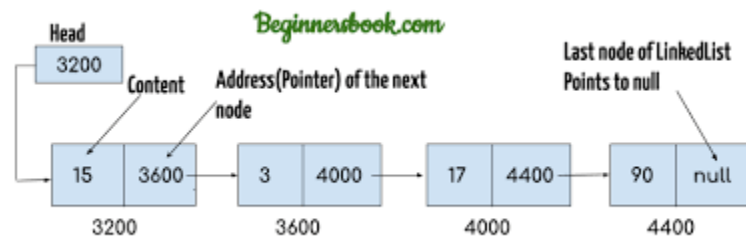
- ii. **Linked List:** Linked list is a linear data structure. It is a collection of nodes stored at non-contiguous memory location.

Each node is divided into two parts:

DataField : Which contains data.

Pointer Field: which contains the address of the next node or adjacent node of the list.

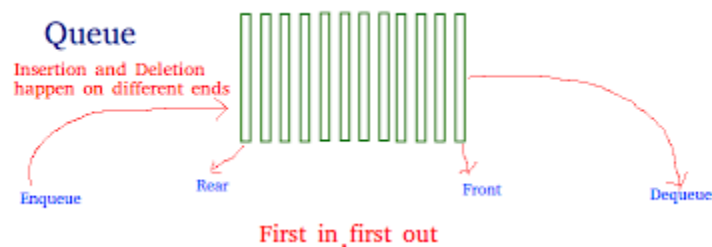
Last node of Linked list points to Null.



Note: In Linked List, nodes are stored at non contiguous memory location means non sequentially but with the help of pointers nodes data are accessed sequentially. so Linked list is a linear data structure.

- iii. **Queue:** Queue is a Linear data structure in which Insertions done at one place called Rear End and Deletions done at another place called Front End.

Queue works on the principle of First in First out (FIFO).

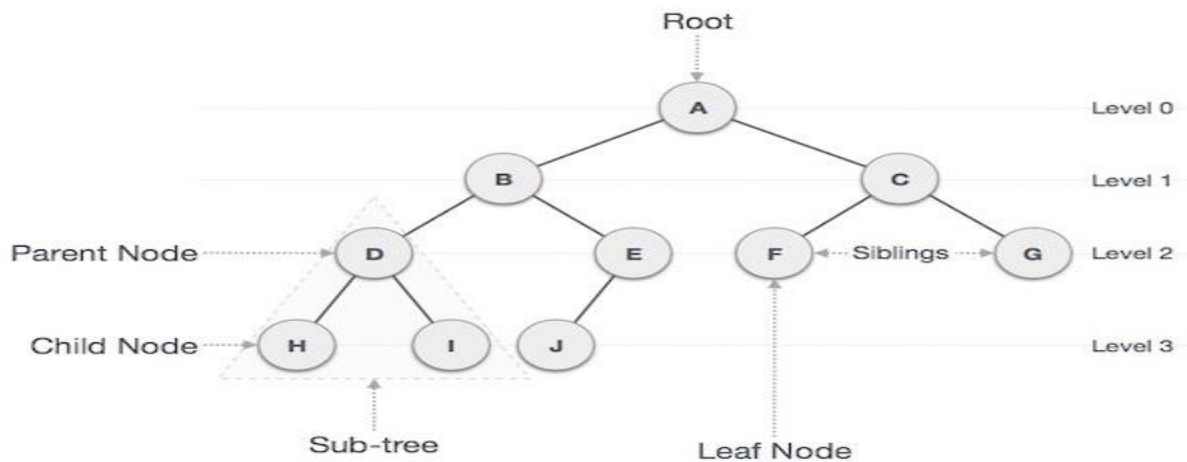


- b. **Non Linear Data Structure:** Non Linear Data Structure does not form a sequence means all its elements are arranged in non- linear order.

Non Linear Data Structures are:

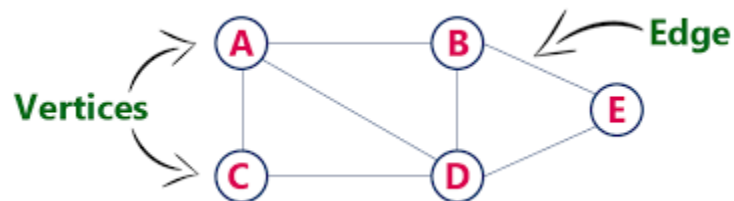
- i. **Tree:** Tree is a non linear data structure which stores data in a hierarchical style in order to maintain hierarchical relationship among data elements.

The topmost node in the hierarchy is called root node and the bottommost node in the hierarchy is called leaf node.



- ii. **Graph:** Graph is a non linear data structure. Graph consists of finite set of vertices and edges which connects a pair of vertices.

$G=\{V,E\}$.



Vertices = { A, B,C,D,E}