**Activity-2**

**Prepare a Note on Linear And Non-Linear Data Structure**

[**Linear Data Structure**](https://www.geeksforgeeks.org/overview-of-data-structures-set-1-linear-data-structures/)**:**

Data structure where data elements are arranged sequentially or linearly where each and every element is attached to its previous and next adjacent is called a linear data structure. In linear data structure, single level is involved. Therefore, we can traverse all the elements in single run only. Linear data structures are easy to implement because computer memory is arranged in a linear way. Its examples are [array](https://www.geeksforgeeks.org/array-data-structure/), [stack](https://www.geeksforgeeks.org/stack-data-structure/), [queue](https://www.geeksforgeeks.org/queue-data-structure/), [linked list](https://www.geeksforgeeks.org/data-structures/linked-list/), etc. 

**1.Array:**

An array consists of data elements of a same data type. For example, if we want to store the roll numbers of 10 students, so instead of creating 10 integer type variables, we will create an array having size 10. Therefore, we can say that an array saves a lot of memory and reduces the length of the code.

**2. Stack:**

It is linear data structure that uses the LIFO (Last In-First Out) rule in which the data added last will be removed first. The addition of data element in a stack is known as a push operation, and the deletion of data element form the list is known as pop operation.

**3. Queue:**

 It is a data structure that uses the FIFO rule (First In-First Out). In this rule, the element which is added first will be removed first. There are two terms used in the queue **front** end and **rear** The insertion operation performed at the back end is known as enqueue, and the deletion operation performed at the front end is known as dequeue.

**4. Linked List:**

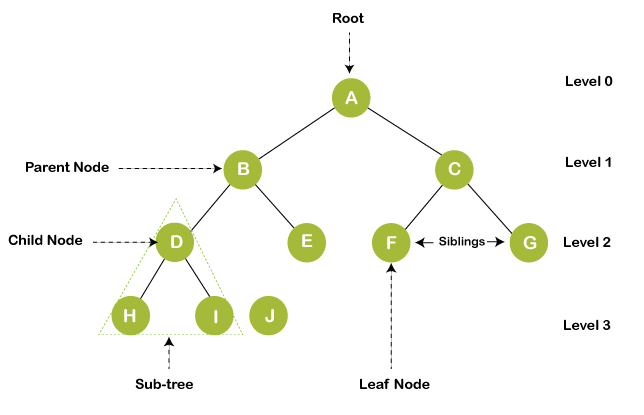
It is a collection of nodes that are made up of two parts, i.e., data element and reference to the next node in the sequence.

[**Non-linear Data Structure**](https://www.geeksforgeeks.org/overview-of-data-structures-set-2-binary-tree-bst-heap-and-hash/)**:**

A non-linear data structure is also another type of data structure in which the data elements are not arranged in a contiguous manner. As the arrangement is nonsequential, so the data elements cannot be traversed or accessed in a single run. In the case of linear data structure, element is connected to two elements (previous and the next element), whereas, in the non-linear data structure, an element can be connected to more than two elements.

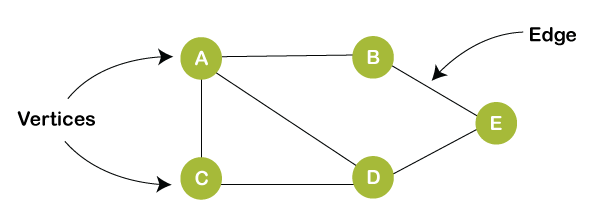
**1.Tree:**

It is a non-linear data structure that consists of various linked nodes. It has a hierarchical tree structure that forms a parent-child relationship. The diagrammatic representation of a **tree** data structure is shown below:



**2. Graph:**

A graph is a non-linear data structure that has a finite number of vertices and edges, and these edges are used to connect the vertices. The vertices are used to store the data elements, while the edges represent the relationship between the vertices. A graph is used in various real-world problems like telephone networks, circuit networks, social networks like LinkedIn, Facebook. In the case of facebook, a single user can be considered as a node, and the connection of a user with others is known as edges.



### Differences between the Linear data structure and non-linear data structure:

|  |  |  |
| --- | --- | --- |
| **Linear Data**  **structure** | | **Non-Linear Data structure** |
| **Basic** | In this structure, the elements are arranged sequentially or linearly and attached to one another. | In this structure, the elements are arranged hierarchically or non-linear manner. | |
| **Types** | Arrays, linked list, stack, queue are the types of a linear data structure. | Trees and graphs are the types of a non-linear data structure. | |
| **implementation** | Due to the linear organization, they are easy to implement. | Due to the non-linear organization, they are difficult to implement. | |
| **Traversal** | As linear data structure is a single level, so it requires a single run to traverse each data item. | The data items in a non-linear data structure cannot be accessed in a single run. It requires multiple runs to be traversed. | |
| **Arrangement** | Each data item is attached to the previous and next items. | Each item is attached to many other items. | |
| **Levels** | This data structure does not contain any hierarchy, and all the data elements are organized in a single level. | In this, the data elements are arranged in multiple levels. | |
| **Memory utilization** | In this, the memory utilization is not efficient. | In this, memory is utilized in a very efficient manner. | |
| **Time complexity** | The time complexity of linear data structure increases with the increase in the input size. | The time complexity of non-linear data structure often remains same with the increase in the input size. | |
| **Applications** | Linear data structures are mainly used for developing the software. | Non-linear data structures are used in **image processing** and **Artificial Intelligence**. | |