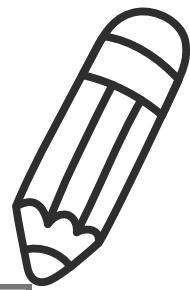




Non Linear

Data Structure



DSA Prep

System Design

Mock Interview



What is Non Linear Data Structure?

Non-linear data structures are those where data items are not arranged in a sequential manner, unlike linear data structures. In these data structures, elements are stored in a hierarchical or a network-based structure that does not follow a sequential order. These data structures allow efficient searching, insertion, and deletion of elements from the structure.

Examples of Non Linear Data Structures:

- Trees
- Graphs, etc.



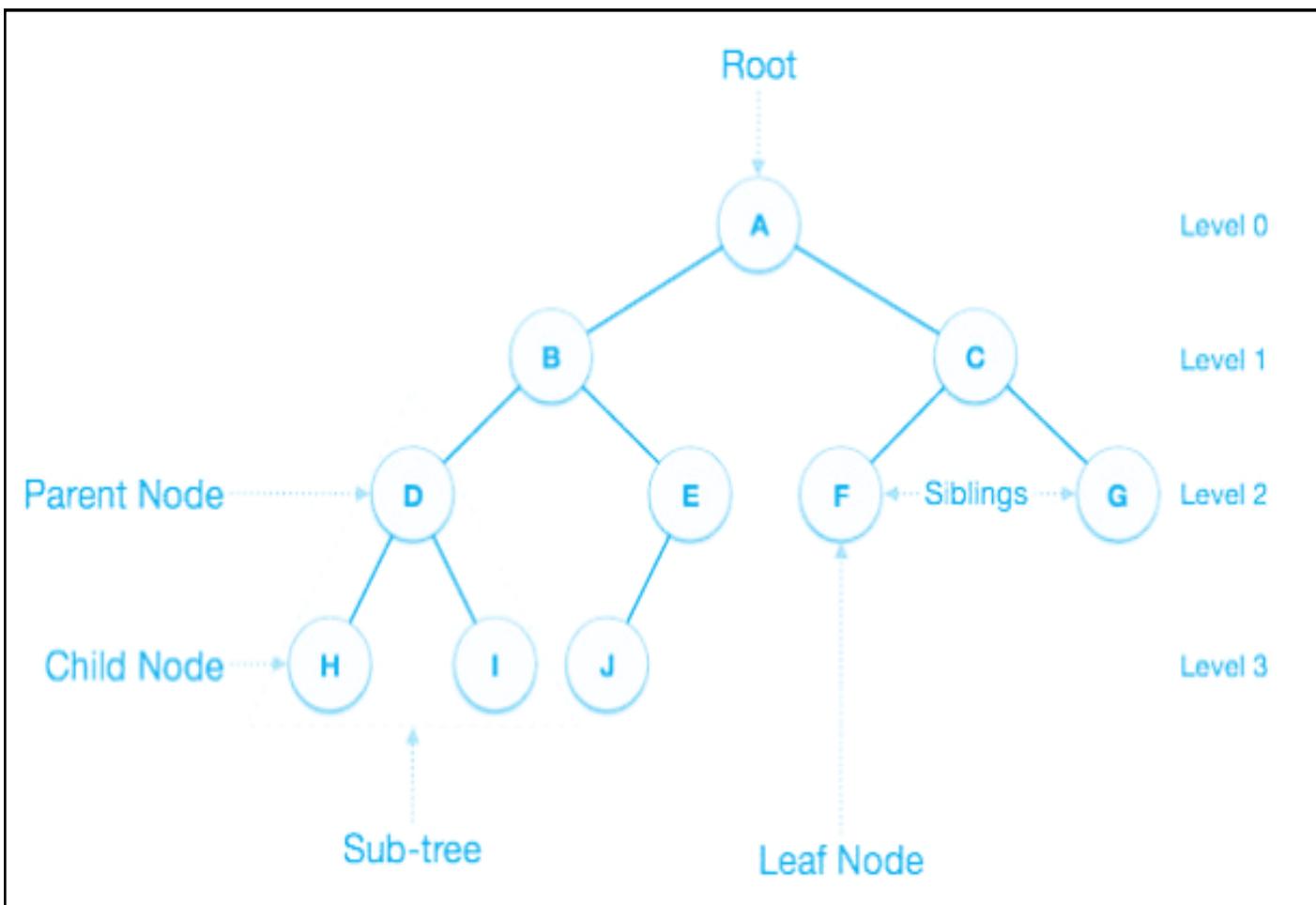
Properties of Non Linear Data Structures

The Non linear data structures have the following properties.

- Non-linear data structures do not follow a sequential order.
- Elements are stored in a hierarchical or a network-based structure.
- These data structures allow efficient searching, insertion, and deletion of elements.
- Non-linear data structures are used to solve complex problems where data cannot be arranged in a linear manner.

Tree Data Structure

A tree is a non linear data structure in which data is stored in a hierarchical structure. It is a collection of nodes connected by edges. Each node has a parent node and zero or more child nodes. The node that is present at the top of the hierarchy is called the root node. Trees are widely used in computer science and are an essential part of many algorithms and data structures.





Terminologies related to Trees

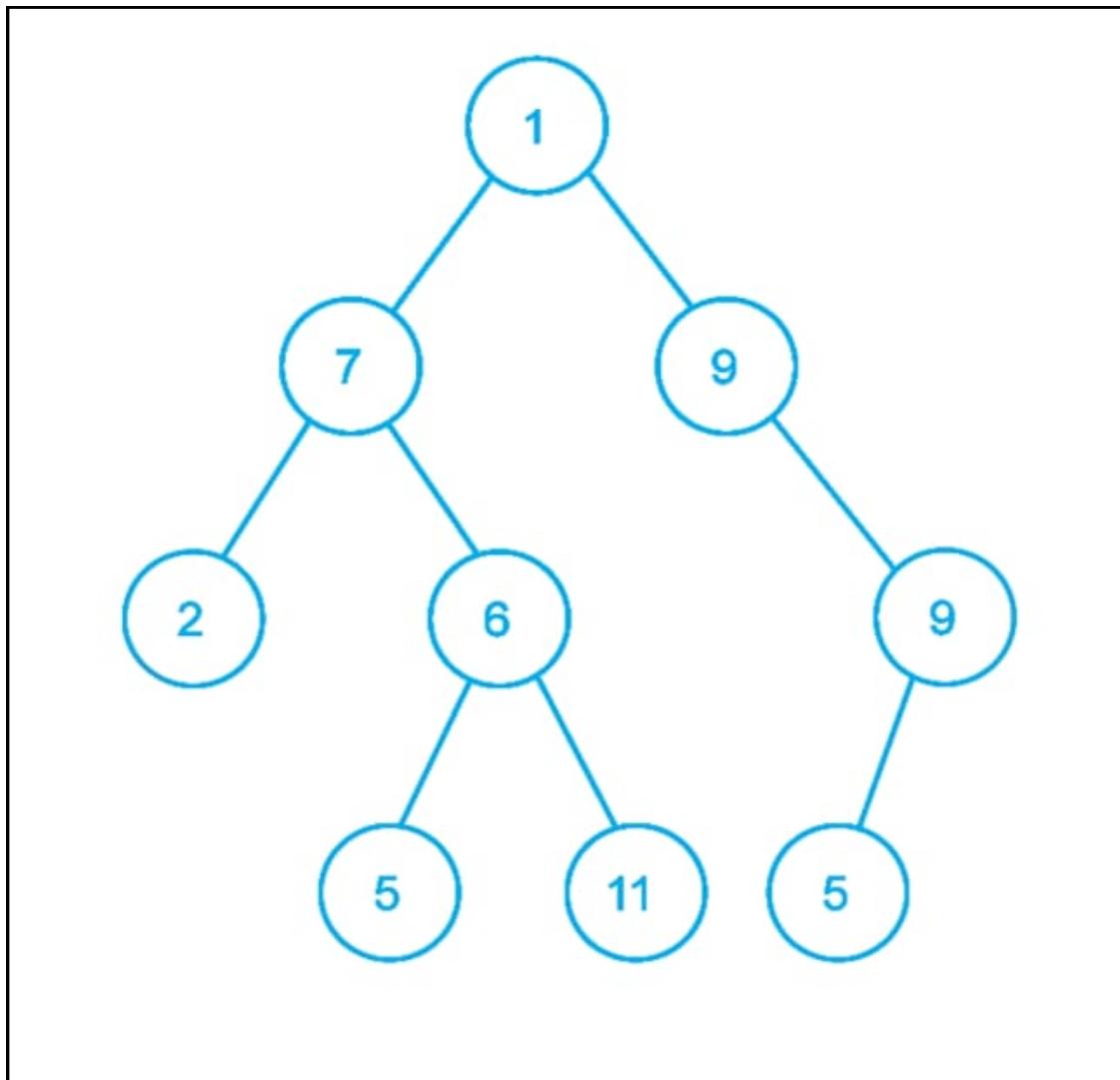
Some terminologies related to trees are given below

- **Node:** A node is a data item that is stored in a tree data structure.
- **Edge:** Two nodes in a tree data structure are connected with the help of an edge.
- **Parent:** A parent in a tree is defined as a node that has one or more child nodes.
- **Child:** A child is a node that is connected to a parent node.
- **Root:** The root node is the topmost node in a tree.
- Leaf: A leaf node in a tree data structure is a node that does not have any child nodes.
- **Height:** The height of a tree is the maximum number of edges from the root node to any leaf node.
- **Depth:** The depth of a node is the number of edges from the root node to that node.

Types of Trees

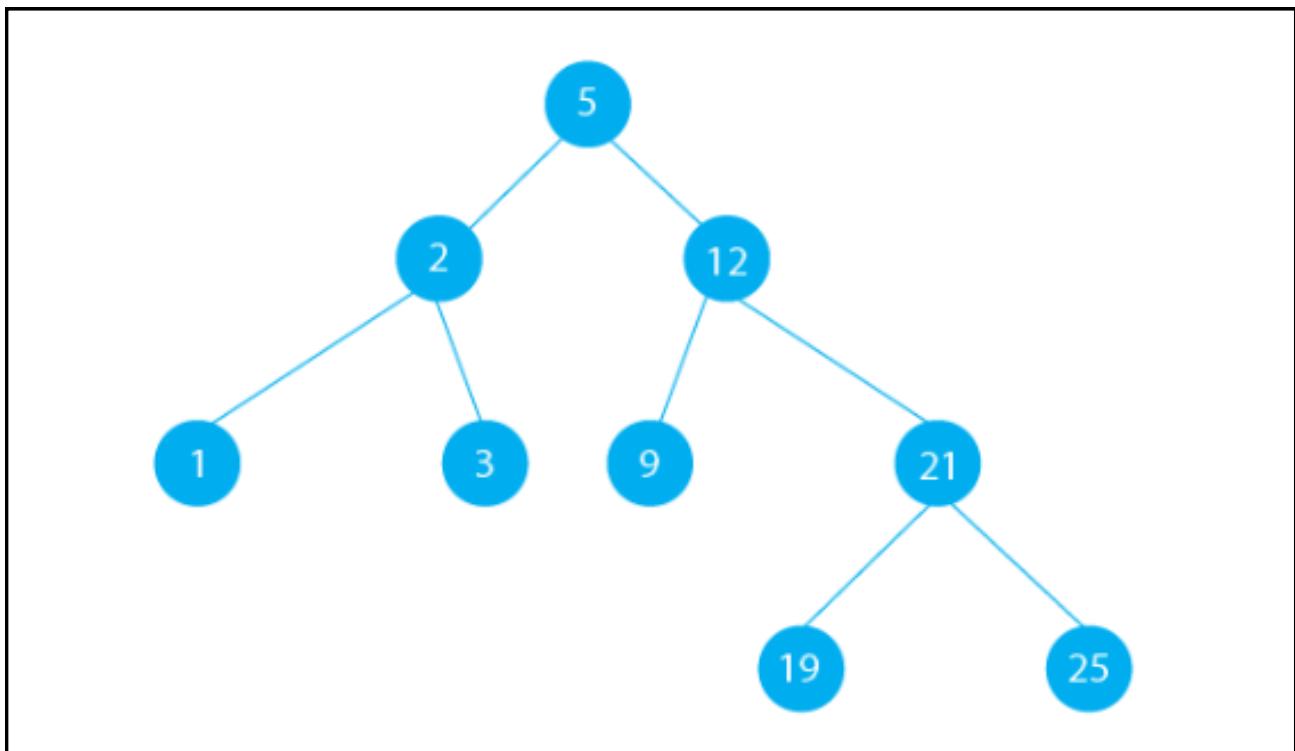
Trees are of the following different types.

- **Binary Tree:** A binary tree is a tree data structure in which each node has at most two child nodes



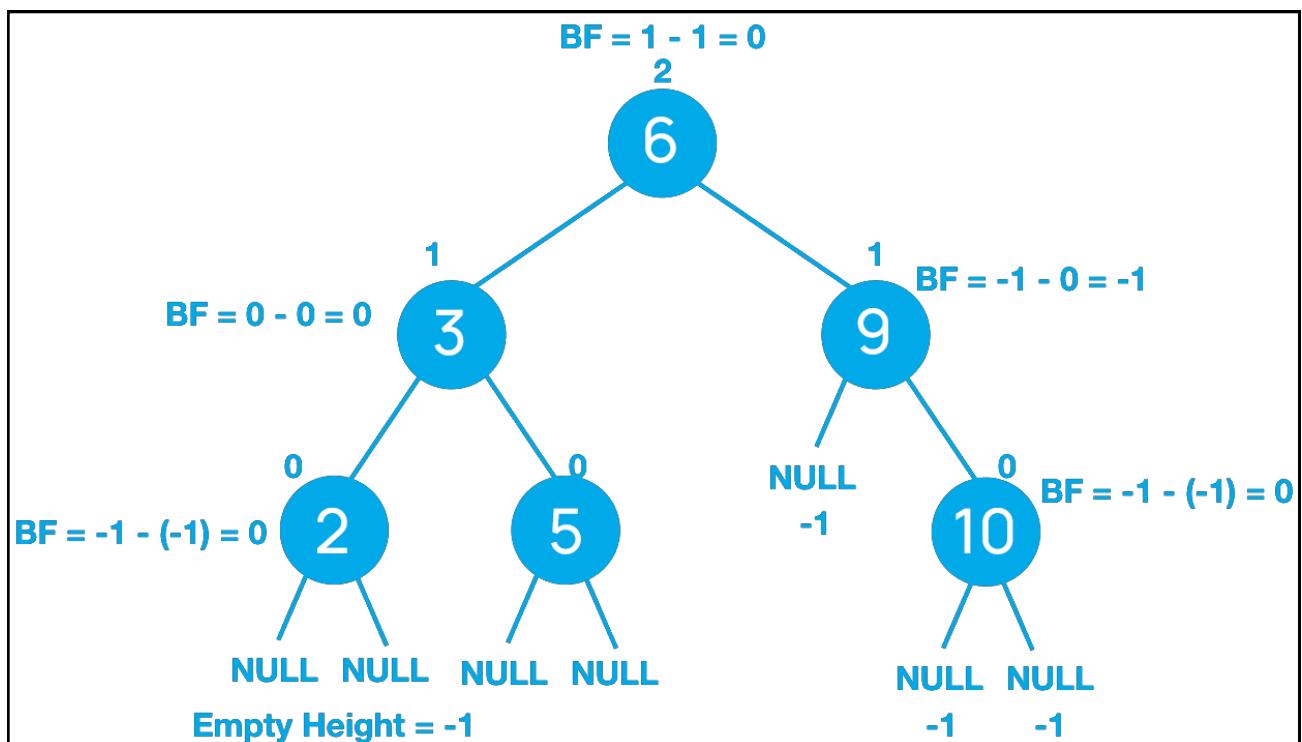
Binary Search Tree

It is a special type of binary tree in which the value of the left node is always smaller than the root node which is always smaller than the right node.



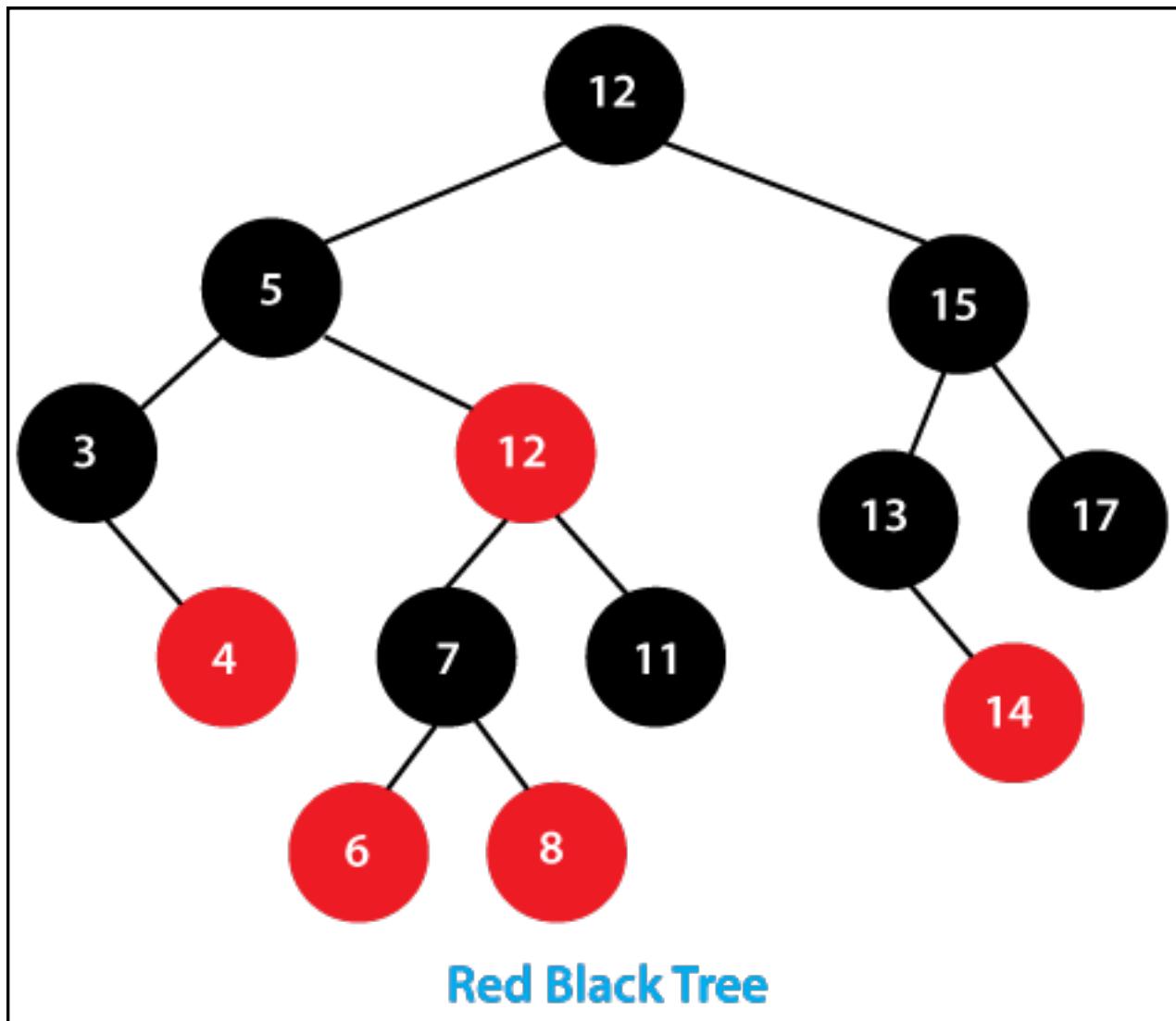
AVL Tree

- An AVL tree is a self-balancing binary search tree. The balancing factor is either 1, -1, or 0.



Red Black Tree

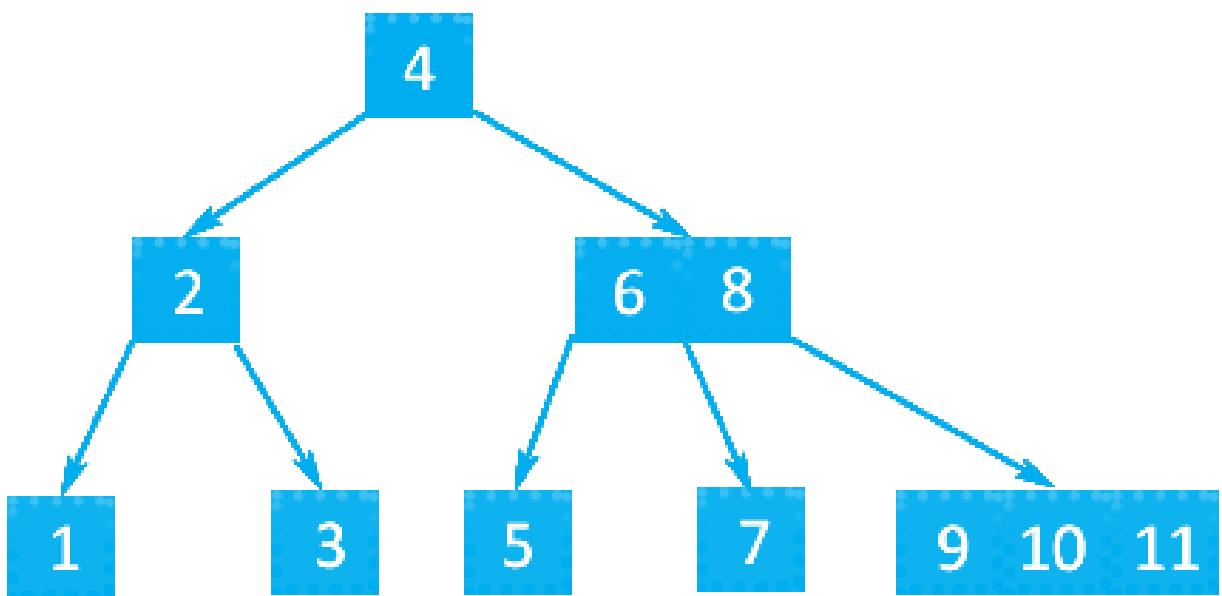
- A red-black tree is a type of binary tree that is self-balanced and each node in it is either colored either red or black.



B-tree

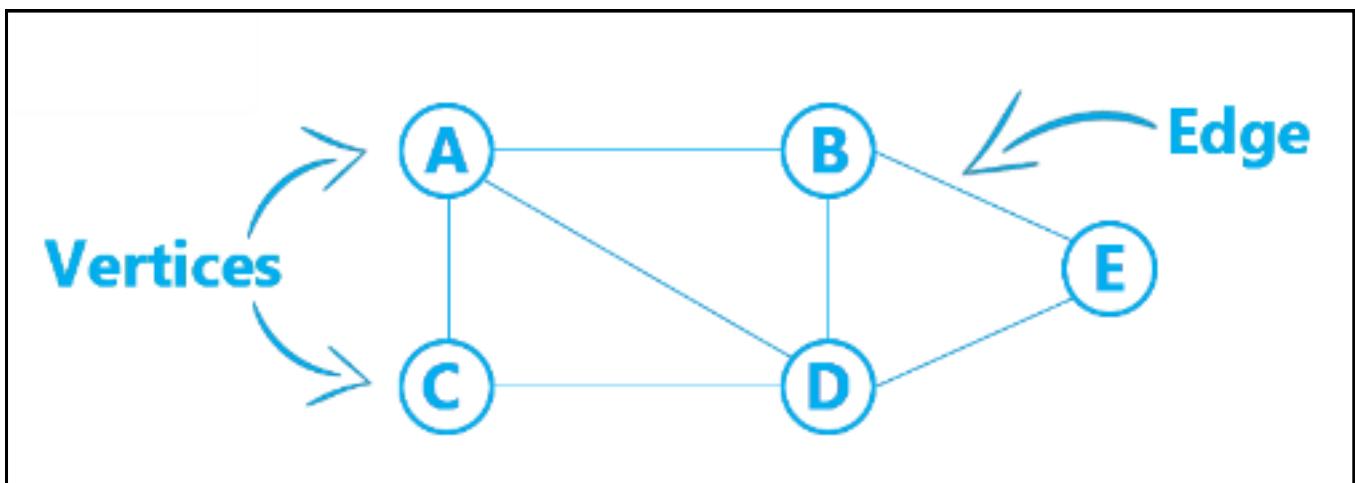
- A B-tree is a self-balancing tree data structure that is commonly used in file systems and DBMS as they facilitate the fast feature of fast searching.

B-TREE



Graph Data Structure

- A graph is a non linear data structure that consists of a set of vertices and a set of edges that connect them together. In a graph, vertices represent entities, while edges represent the relationships between them. Graphs are used to represent complex relationships between entities and are widely used in computer science.





Terminologies Related to Graphs

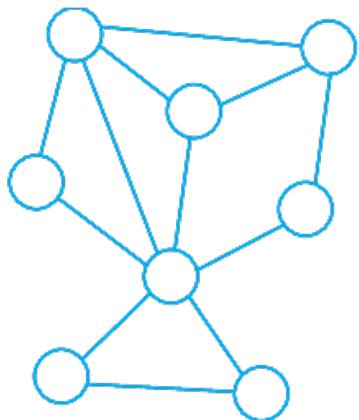
Here are some terminologies related to graphs.

- **Vertex:** A vertex is a data item that is stored in a graph data structure.
- **Edge:** An edge is a connection between two vertices in a graph.
- **Degree:** The degree of a vertex in a graph data structure is defined as the number of edges connected to it.
- **Weight:** The weight of an edge is a numerical value that is assigned to the edge to represent the cost or distance between the vertices.
- **Path:** A path is a sequence of edges that connects two vertices in a graph.
- **Cycle:** A cycle is a path in a graph that starts and ends at the same vertex.

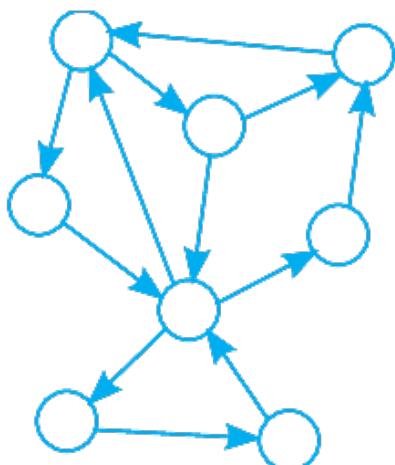
Types of Graphs

Graphs are of the following different types.

- **Undirected Graph:** In an undirected graph, edges do not have a direction. That is, the edge connecting vertex A to vertex B is the same as the edge connecting vertex B to vertex A.

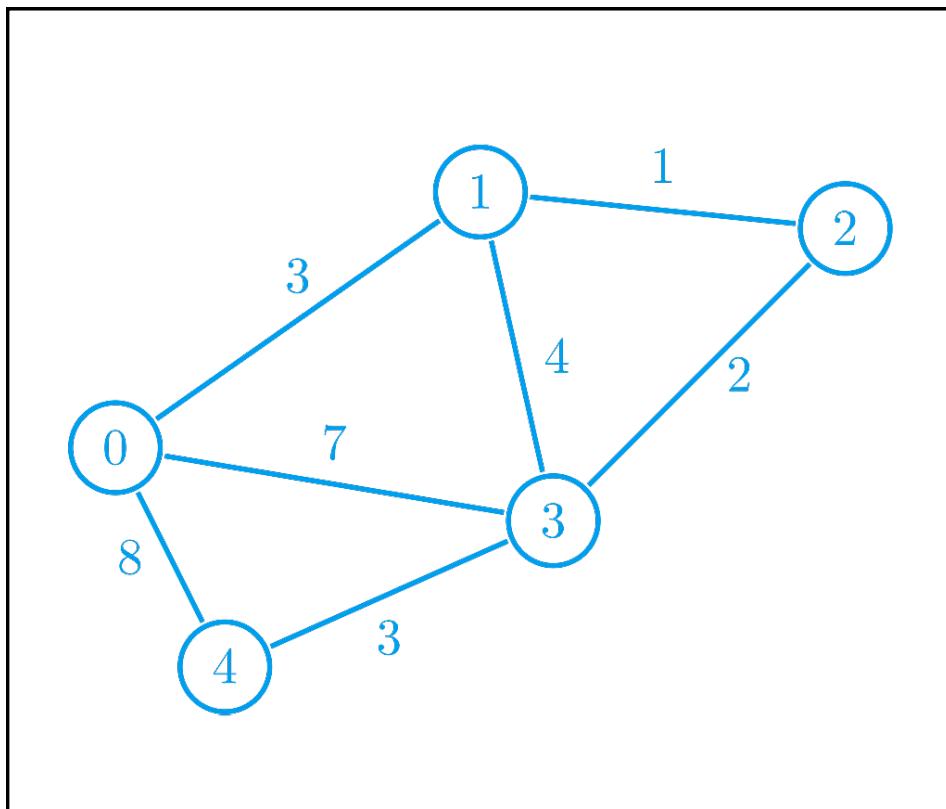


- **Directed Graph:** In the case of a directed graph, the edges have a direction that means, the edge connecting vertex A to vertex B is different from the edge connecting vertex B to vertex A.



Weighted Graph

In a weighted graph, edges have weights that represent the cost or distance between the vertices..





Conclusion

- Non linear data structures are essential as they are used to solve complex problems where data cannot be arranged in a linear manner.
- Tree and graph data structures are examples of non-linear data structures, and they are used to represent hierarchical and network-based relationships between entities, respectively.



Frequently asked Questions

Here are some frequently asked questions related to non linear data structures.

Ques 1. What are the advantages of using a tree data structure?

Ans. The tree data structure is a non linear data structure that has several advantages, including efficient searching and insertion, easy traversal, and natural representation of hierarchical relationships.

Ques 2. What is a tree traversal?

Ans. Tree traversal is the process of visiting each node in a tree exactly once in a specific order.

Ques 3. What are the types of tree traversals?

Ans. The three main types of tree traversals are in-order traversal, pre-order traversal, and post-order traversal.

Ques 4. What are different ways in which we can traverse a graph data structure?

Ans. The two ways for traversing the graphs are

- DFS (Depth First Search)
- BFS (Breadth First Search)