2a) Explain about non linear data structure.

The data structure in which the data elements are randomly arranged. The elements are non-arranged sequentially. The data elements are present at different levels. In Non-linear data structures, there are different paths for an element to reach the other element. The data elements in the non-linear data structures are connected to one or more elements. There are two types of non-linear data structures. They are:

- **Tree Data Structure**
- **Graph Data Structure**

Tree Data Structure:

Tree data structures are completely different from the arrays, stacks, queues and linked lists. Tree data structures are hierarchic. The tree data structure collects the nodes together to depict and stimulate the sequence. Tree data structure does not store the data sequentially. It stores the data on multiple levels. The top node of the Tree Data Structure is known as the Root Node. Any type of data can be stored in the root node. Each node shall definitely contain the data. The branches in the Tree Data Structure are known as the children.

The different parts of the Tree Data Structure are:

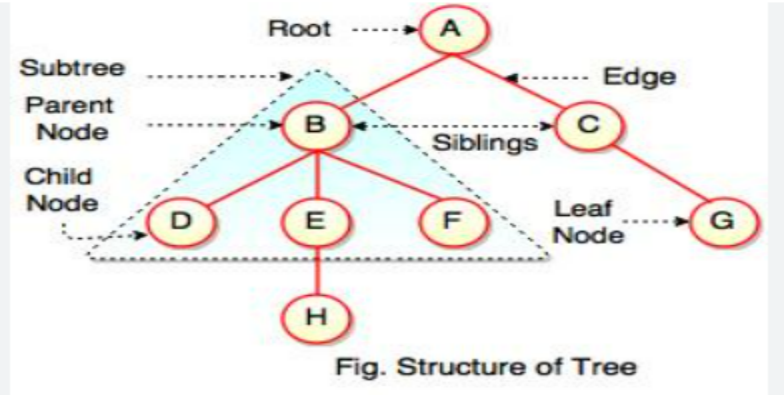
1. Root Node
2. Child Node
3. Edge
4. Siblings
5. Leaf Node
6. Internal Nodes
7. Height of the tree
8. Degree of the Node
9. 

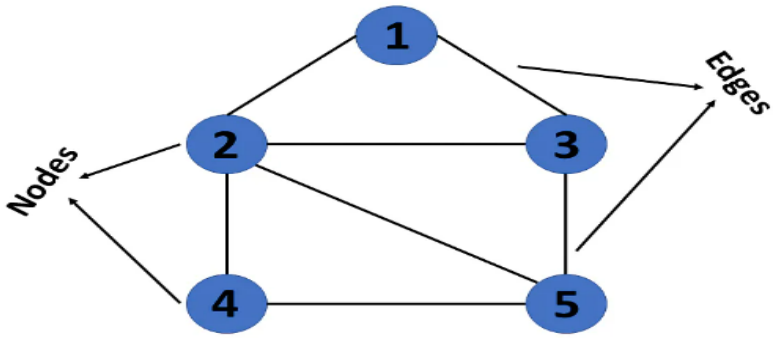
Fig. Structure of Tree

Graph Data Structure

In Graph Data Structure, one node is simply connected to the other node through the edge of the graph. The Graph Data Structure obviously uses Non-linear data structures which are not sequentially arranged. The graph data structures consist of edges and nodes represented by E and V, respectively. Graph Data Structures do not have root nodes. It does not have a standard order of arranging the data. Every tree is also known as the graph with n-1 edges where 'n' represents the total number of vertices in the graph. There are various categories in the graphs such as undirected, unweighted, directed and weighted.

The different parts of the graph are as follows.

1. Vertex
2. Edges
3. Directed Edge
4. Undirected Edge
5. Weighted Edge
6. Degree
7. Indegree
8. Outdegree



Representation of Graphs in Data Structures

Graphs in data structures are used to represent the relationships between objects. Every graph consists of a set of points known as vertices or nodes connected by lines known as edges. The vertices in a network represent entities.

The most frequent graph representations are the two that follow:

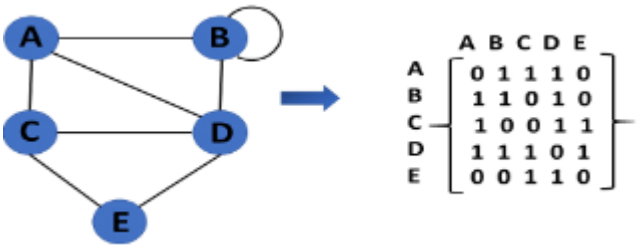
- Adjacency matrix
- Adjacency list

You'll look at these two representations of graphs in data structures in more detail:

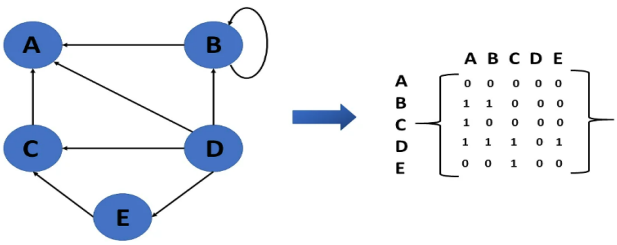
Adjacency Matrix

- A sequential representation is an adjacency matrix.
- It's used to show which nodes are next to one another. I.e., is there any connection between nodes in a graph?
- You create an MXM matrix G for this representation. If an edge exists between vertex a and vertex b, the corresponding element of G, gi,j = 1, otherwise gi,j = 0.
- If there is a weighted graph, you can record the edge's weight instead of 1s and 0s.

Undirected Graph Representation



Directed Graph Representation



Weighted Undirected Graph Representation

Weight or cost is indicated at the graph's edge, a weighted graph representing these values in the matrix.