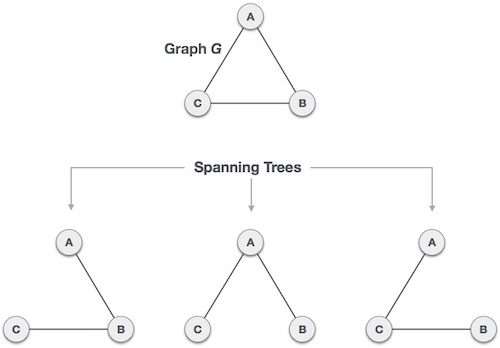
**Day 11**

**1) Spanning trees-**

A spanning tree is a subset of Graph G, which has all the vertices covered with minimum possible number of edges. Hence, a spanning tree does not have cycles and it cannot be disconnected.

By this definition, we can draw a conclusion that every connected and undirected Graph G has at least one spanning tree. A disconnected graph does not have any spanning tree, as it cannot be spanned to all its vertices.



A complete undirected graph can have maximum n^(n-2) number of spanning trees, where n is the number of nodes. In the above addressed example, n is 3, hence 3^(3−2) = 3 spanning trees are possible.

**1.1) General Properties of Spanning Tree-**

We now understand that one graph can have more than one spanning tree. Following are a few properties of the spanning tree connected to graph G −

a) A connected graph G can have more than one spanning tree.

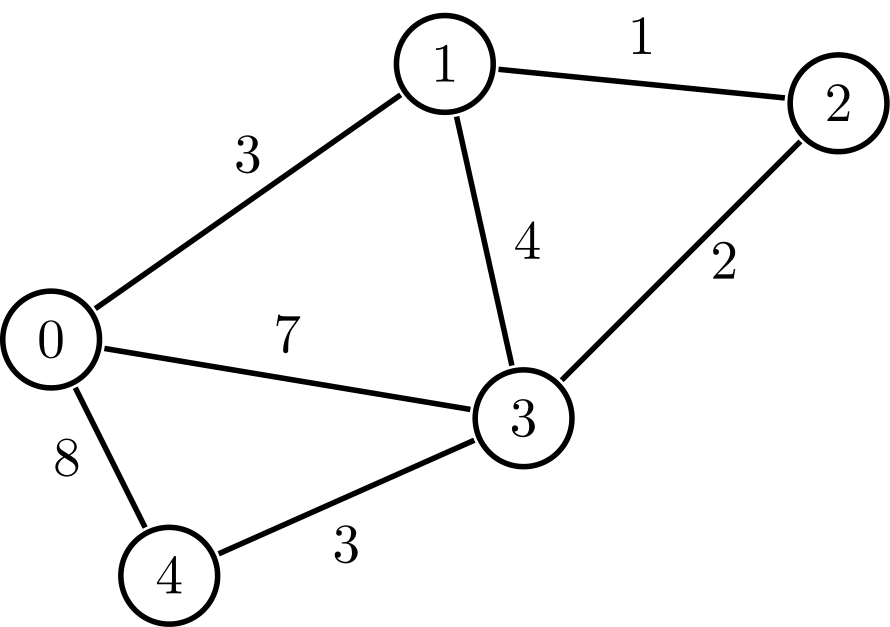
b) All possible spanning trees of graph G, have the same number of edges and vertices.

c) The spanning tree does not have any cycle (loops).

d) Removing one edge from the spanning tree will make the graph disconnected, i.e. the spanning tree is minimally connected.

e) Adding one edge to the spanning tree will create a circuit or loop, i.e. the spanning tree is maximally acyclic.

**1.2- Weighted graphs-**



A graph is a set of nodes and edges, where each edge represents a connection between two nodes. A weighted graph is a graph where each edge has a numerical value called weight. In real-world situations, this weight can be measured as distance, congestion, traffic load or any arbitrary value denoted to the edges.

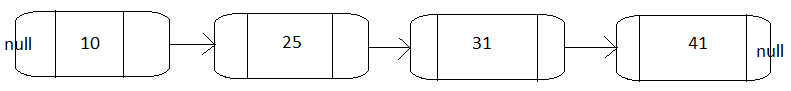
**1.3) Minimum Spanning Tree (MST)-**

In a weighted graph, a minimum spanning tree is a spanning tree that has minimum weight than all other spanning trees of the same graph.

**3) Types of linked lists-**

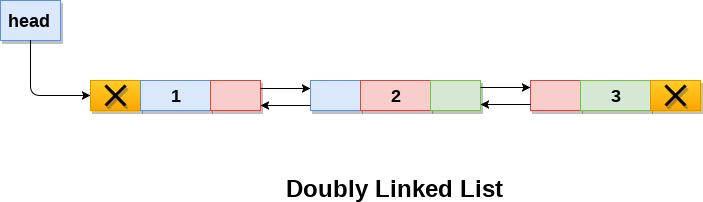
**3.1) Normal linked list -**

Linked list is a linear data structure which is used to maintain a list in the memory. It can be seen as the collection of nodes stored at non-contiguous memory locations. Each node of the list contains a pointer to its next node.



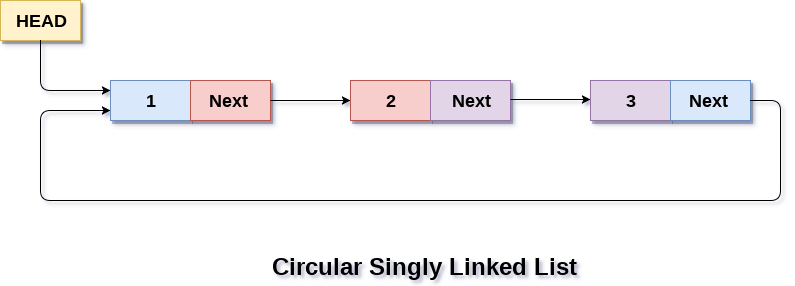
**3.2) Doubly linked list -**

Doubly linked list is a linked list in which a node contains a pointer to the previous as well as the next node in the sequence. Therefore, in a doubly linked list, a node consists of three parts: node data, pointer to the next node in sequence (next pointer) , pointer to the previous node (previous pointer).



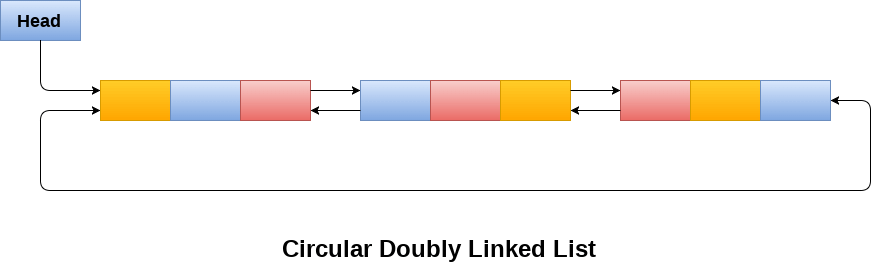
**3.3) Circular Singly linked list-**

In a circular Singly linked list, the last node of the list contains a pointer to the first node of the list. We can have circular singly linked list as well as circular doubly linked list.



**3.4) Circular Doubly linked list-**

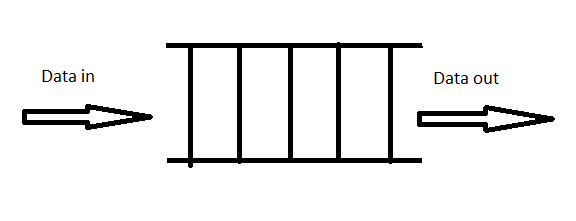
Circular doubly linked list contains pointers to its previous node as well as the next node. Circular doubly linked list doesn't contain NULL in any of the node. The last node of the list contains the address of the first node of the list. The first node of the list also contain address of the last node in its previous pointer.



**4) Queues-**

**4.1) Normal Queue-**

Queue is a linear list in which elements can be inserted only at one end called rear and deleted only at the other end called front.



**4.2) Circular Queue-**

In a normal Queue, we can insert elements until queue becomes full. But once queue becomes full, we can not insert the next element even if there is a space in front of queue. In circular queue, last position is connected back to the first position to make a circle. It is also called ‘Ring Buffer’. Like normal queue, circular queue also works on FIFO (First In First Out) principle.

