to value stored in a large database that is stored on a disk is a very time consuming process.

searching an un-indexed and unsorted database containing n key values needs O(n) running time.

6). B + Tree :-

B+tree is an extension of B tree which allows efficient insertion, deletion and search operations.

The leaf nodes of B+ tree are linked together in form of the singly linked list tomake search queries more efficient.

Advantages of B+ tree :-

1). Records can be fetched in equal number of disk accesses.

2) Height of tree remains balanced and less as compare to B tree.

3) we can access data stored in B+ tree sequentially as well as directly.

4) keys are used for indexing.

Graph :-

A graph can be defined as group of vertices and edges that are used to connect these vertices.

Defination :-

A graph G can be defined as an ordered set G(V,E) where V(G) represents set of vertices and E(G) represents set of edges.

which are used to connect these vertices.

Directed and Undirected Graph :-

A graph can be directed or undirected. However, in an undirected graph, edges are not associated with directions with them.

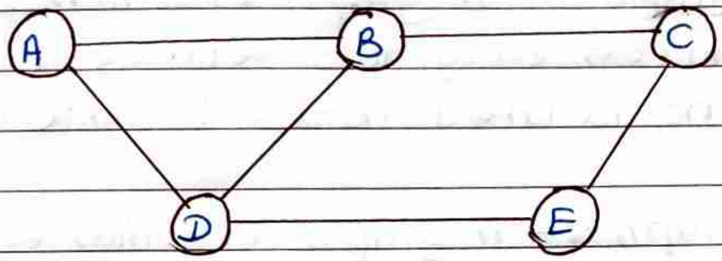


Fig : undirected graph

As above figure edges are not attached with any of the directions.

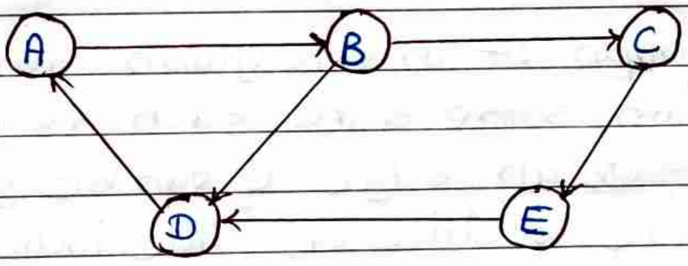


Fig: directed graph.

In above figure, directed graph edges form an ordered pair.

Graph Terminology :-

1). Path :- A path can be defined as sequence of nodes that are followed in order to reach some terminal node v from initial node U.

2). closed path :- A path will be called as closed if initial node is same as terminal node. $V_0 = V_N$

3). Simple path :- If all nodes of graph are distinct with an exception $V_0 = V_N$, then such path P is called as closed simple path.

4). Cycle :- A cycle is a path which has no repeated edges or vertices except first and last vertices.

5). Connected graph :- A graph in which some path exists between every two vertices (u,v) in V. There are no isolated nodes in connected graph.

6). Complete graph :- A graph in which every node is connected with all other nodes. A complete graph contain $\frac{n(n-1)}{2}$ edges where n is number of nodes in graph.

7). Weighted graph :- In this graph each node is assigned with some data such as length or width. The weight of an edge e can be given as w(e) which must be positive (+) value indicating cost of traversing edge.

8). Diagraph :- A diagraph is directed graph in which each edge is associated with some direction and traversing can be done only in specified direction.

9). Loop :- An edge that is associated with the similar end points can be called as loop.

10). Adjacent Nodes :- If two nodes u and v are connected via an edge e, then nodes u and v are called as neighbours or adjacent nodes.

11). Degree of a Node :- A degree of a node is a number of edges that are connected with that node. A node with degree 0 is called isolated.

Graph Representation :-

We simply mean, technique which is to be used to in order to store some graph into the computers memory.

(1). Sequential Representation :- In this we use adjancy matrix to store mapping represented by vertices and edges. A graph having n vertices, will have a dimension n x n.

An entry M_{ij} in adjacency matrix representation of an ~~mat~~ undirected graph G will be 1 if there exists an edge between v_i and v_j.

An undirected graph and its adjacency matrix representation is shown in following :

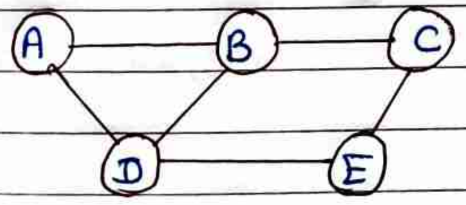


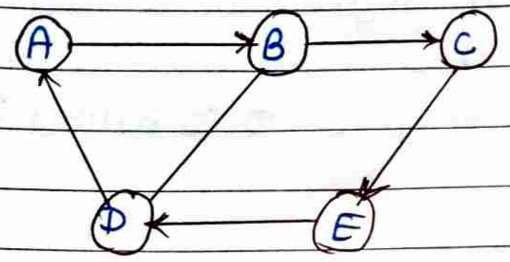
fig : Undirected graph

$$\begin{array}{c|ccccc|} & A & B & C & D & E \\ A & 0 & 1 & 0 & 1 & 0 \\ B & 1 & 0 & 1 & 1 & 0 \\ C & 0 & 1 & 0 & 0 & 1 \\ D & 1 & 1 & 0 & 0 & 1 \\ E & 0 & 0 & 1 & 1 & 0 \end{array}$$

Fig: Adjacency matrix

In above figure, we can see mapping among vertices (A, B, C, D, E) is represented by using adjucency matrix which is also shown in fig.

A directed graph and its adjaceny matrix representation is shown in figure :

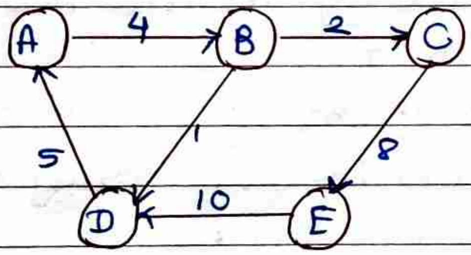


$$\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & 0 & 1 & 0 & 0 & 0 \\ B & 0 & 0 & 1 & 1 & 0 \\ C & 0 & 0 & 0 & 0 & 1 \\ D & 1 & 0 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 1 & 0 \end{array}$$

Fig : Directed Graph

fig: Adjacency matrix

Representation of weighted directed graph is different Instead of filling entry by 1, non zero entries of adjacency matrix are represented by weight of respective edges.

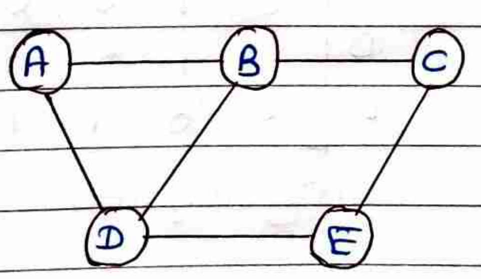


$$\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & 0 & 4 & 0 & 0 & 0 \\ B & 0 & 0 & 2 & 1 & 0 \\ C & 0 & 0 & 0 & 0 & 8 \\ D & 5 & 0 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 10 & 0 \end{array}$$

Fig : weighted directed graph

fig: Adjancy matrix

(2) linked representation :-



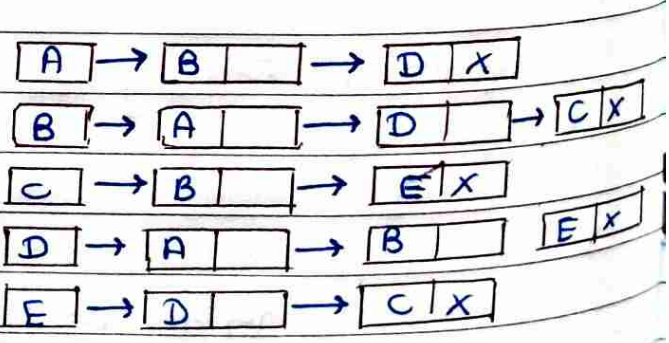


Fig : undirected graph

Fig : Adjacency list.

An adjacency list is maintained for each node present in graph which stores node value and a pointer to next adjacent node to respective node.

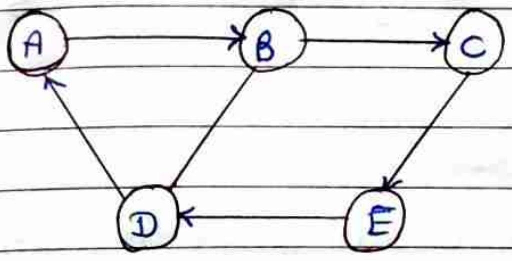


Fig : Directed graph

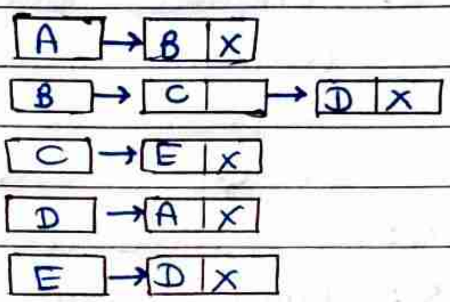


Fig: Adjancy list

In directed graph, sum of lengths of all the adjancy lists is equal to the number of edges present in the graph.

Graph Traversal Algorithm :-

In this tutorial we will learn all techniques by using which, we can traverse all the vertices of the graph. Traversing means examining all nodes and vertices of graph. There are two standard methods by using which, we can traverse graphs.

- Breadth first search
- Depth first search

1. Breadth first search (BFS) algorithm :-

Breadth first search is a graph traversal algorithm that starts traversing graph from root node and explorer all the neighbouring nodes.

Then, it selects nearest node and explore all unexplored nodes. The algorithm follows same process for each of nearest node until it finds goal.