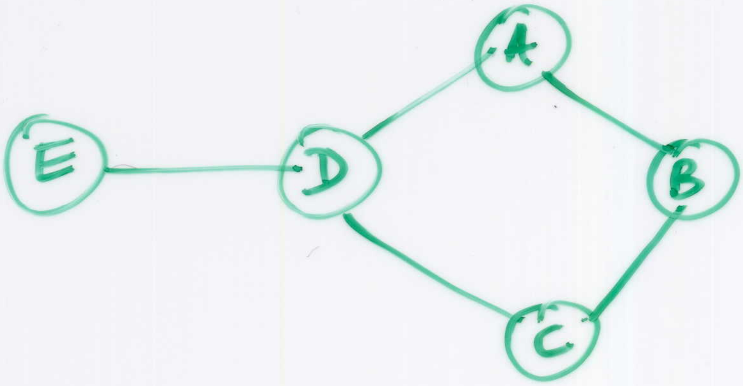
Adjacency matrix of an undirected graph:-



Adjacency matrix:-

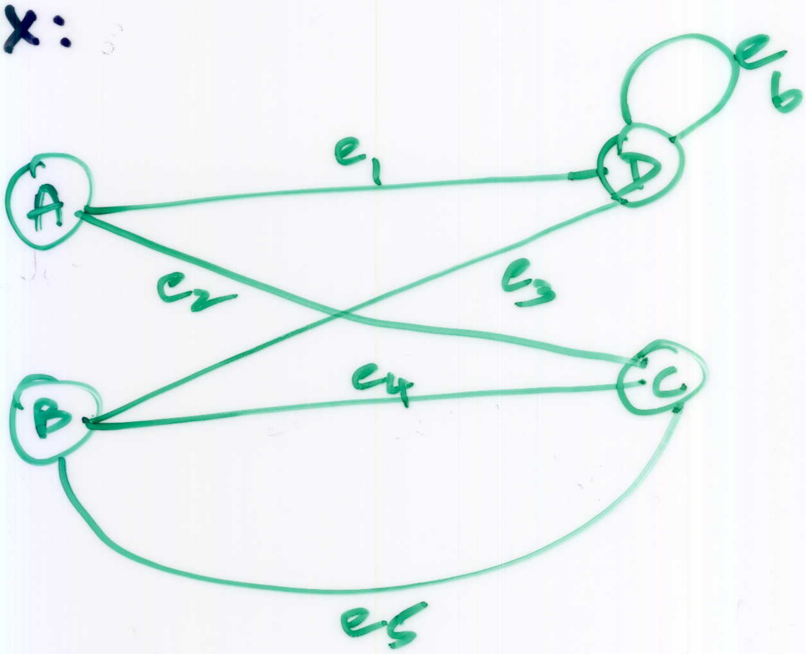
$$
\begin{array}{c|ccccc}
 & A & B & C & D & E \\
\hline
A & 0 & 1 & 0 & 1 & 0 \\
B & 1 & 0 & 1 & 0 & 0 \\
C & 0 & 1 & 0 & 1 & 0 \\
D & 1 & 0 & 1 & 0 & 1 \\
E & 0 & 0 & 0 & 1 & 0
\end{array}
$$

$a_{ij} = 1$ if there is an edge from v_i to v_j

$= 0$ otherwise

Multi-graph :- a graph which has either a self-loop / parallel edges or both

Ex:

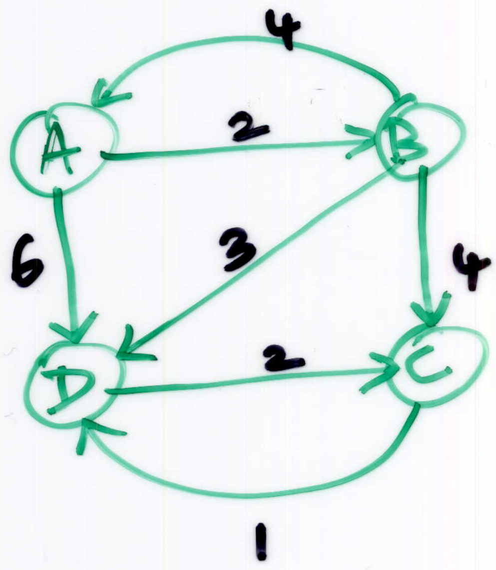


Adjacency matrix of a multi-graph

a_{ij} = no. of edges bet^n 2 vertices v_i and v_j

= 0 no edge bet^n v_i and v_j

Directed Graph:-



Adjacency matrix:-

$$a_{ij} = \text{weight of the edge bet}^n\ v_i \text{ and } v_j$$
$$= 0 \text{ otherwise}$$

$$\begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 0 & 2 & 0 & 6 \\ 4 & 0 & 4 & 3 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 \end{bmatrix} \end{array}$$

Traversing a graph:-

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

BFS:-

From each vertex v, that we visit, search as broadly as possible by next visiting all the vertices adjacent to v.

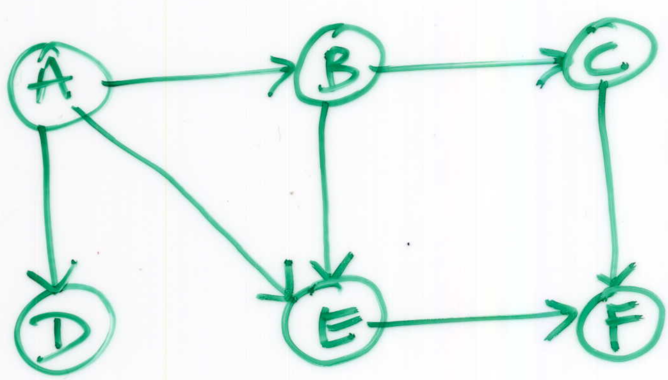
Same as level by level traversal of a tree.

DFS:- Same as traversing a tree by following a path from root to a leaf, then another path from root to a leaf and so on

Breadth - First search

- uses a queue traversing all nodes of the graph.
- Take any node as starting node. Visit it
- Visit all nodes adjacent to the starting node. and so on.
- Maintain the status of visited nodes in an array.
- Every node should be traversed once.

Ex:



A as starting node

BFS Traversal:- A, B, D, E, C, F