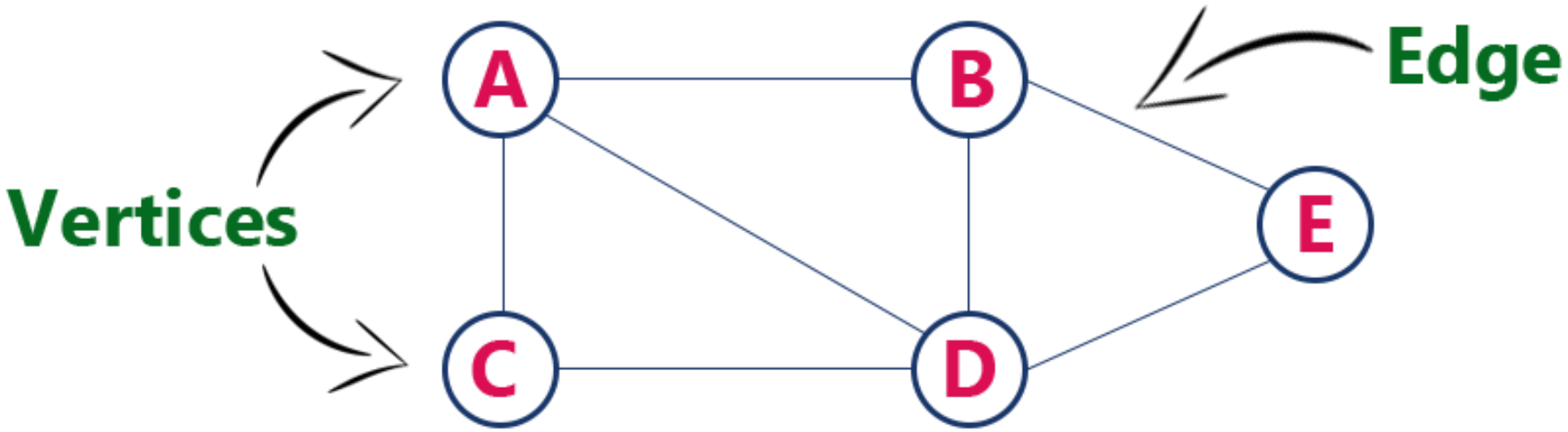
Graph

Graph

- Graph is a non linear data structure.
- Graph is a collection of vertices(nodes) and arcs(edges) which connects vertices in the graph.
- Graph is a collection of nodes and edges which connects nodes in the graph.
- it contains a set of points known as nodes (or vertices) and set of links known as edges (or Arcs) which connects the vertices.
- A graph **G** is represented as **G = (V , E)**, where **V is set of vertices** and **E is set of edges**.

Example: how many Vertices and edges?



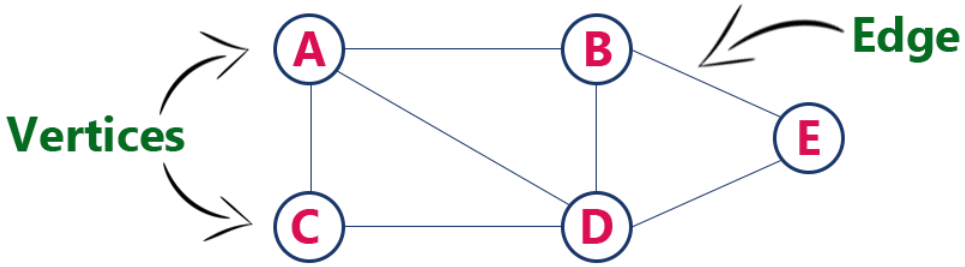
This graph G can be defined as G = (V , E)
Where V = {A,B,C,D,E} and E = {(A,B),(A,C)(A,D),(B,D),(C,D),(B,E),(E,D)}.

Graph theory terminology:

- Vertex
- Edge
- Undirected Graph
- Directed Graph
- Mixed Graph
- Adjacent
- Path
- Outgoing edge
- Incoming edge
- Degree
- InDegree
- OutDegree

- **Vertex**

A individual data element of a graph is called as Vertex. **Vertex** is also known as **node**. In above example graph, A, B, C, D & E are known as vertices.



- **Edge**

An edge is a connecting link between two vertices. **Edge** is also known as **Arc**.

An edge is represented as (startingVertex, endingVertex).

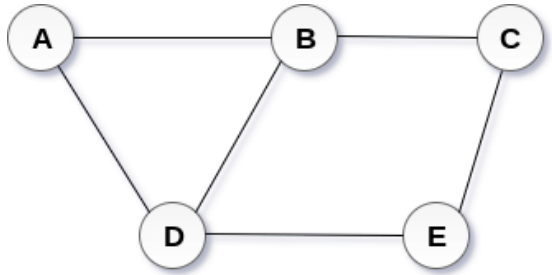
Edge can be directed and undirected

edge

- An undirected edge is a bidirectional edge. If there is a undirected edge between vertices A and B then edge (A , B) is equal to edge (B , A).
- A directed edge is a unidirectional edge. If there is a directed edge between vertices A and B then edge (A , B) is not equal to edge (B , A).

- **Undirected Graph**

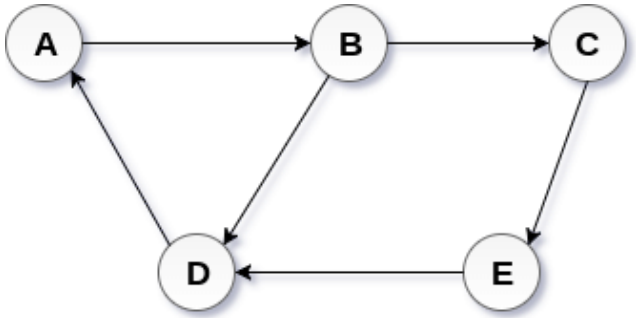
A graph with only undirected edges(bidirectional edge) is said to be undirected graph.



Undirected Graph

- **Directed Graph**

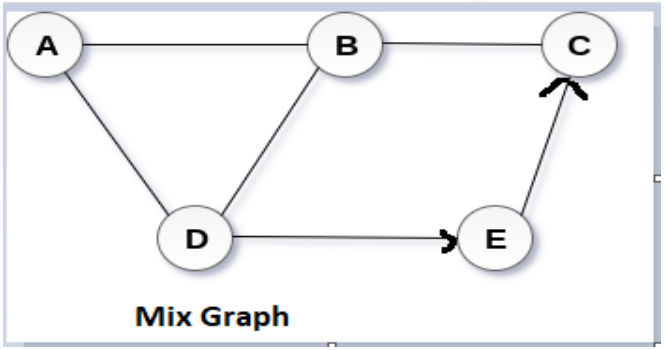
A graph with only directed edges is said to be directed graph.



Directed Graph

- **Mixed Graph**

A graph with undirected and directed edges is said to be mixed graph.



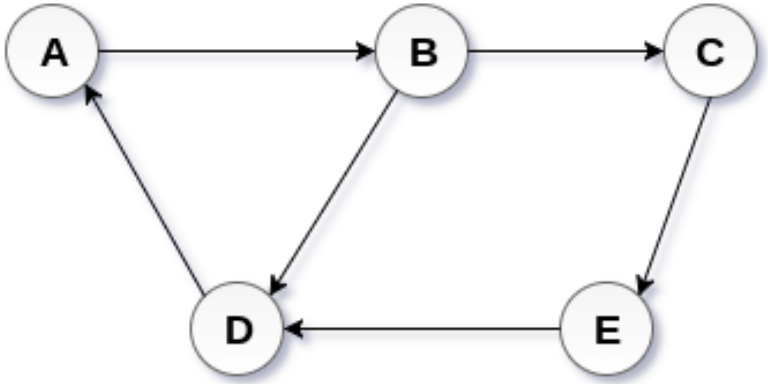
- **Adjacent**

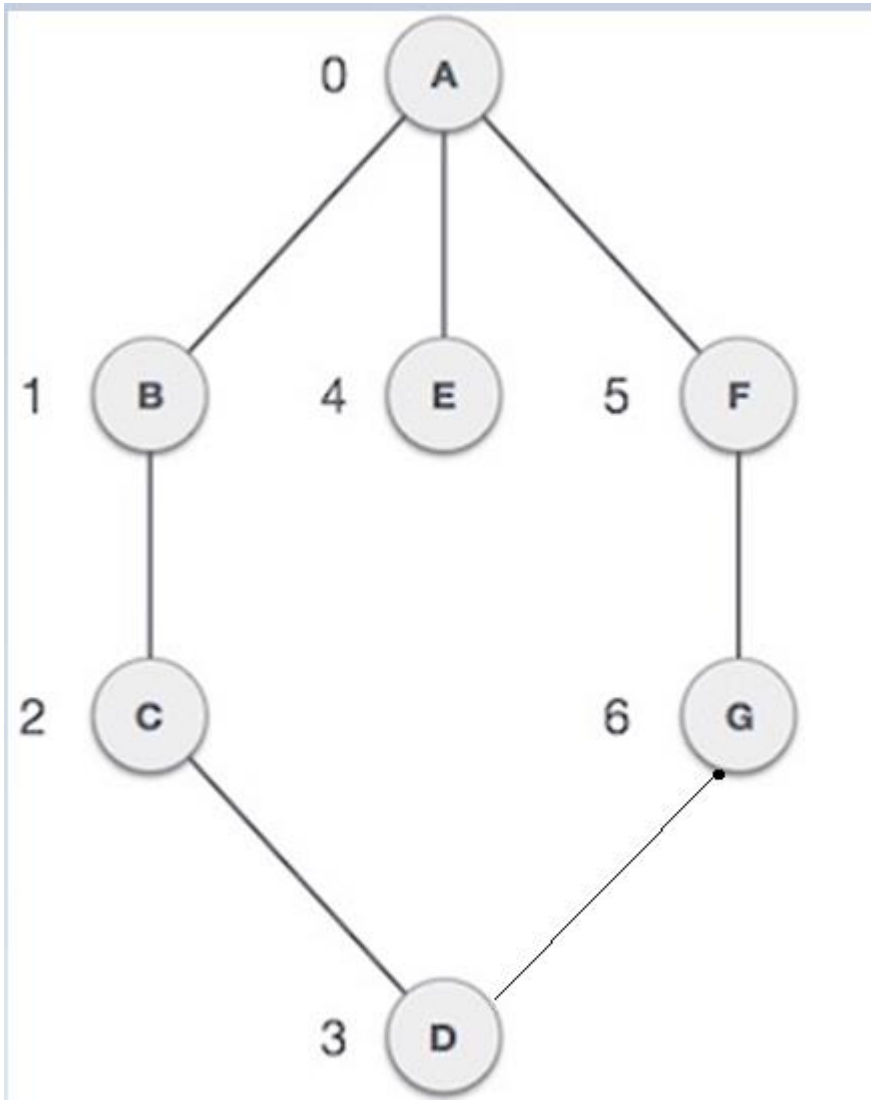
Two node or vertices are adjacent if they are connected to each other through an edge.

Ex:-

G=(A,B) A is adjacent to B

Path – Path represents a sequence of edges between the two vertices.





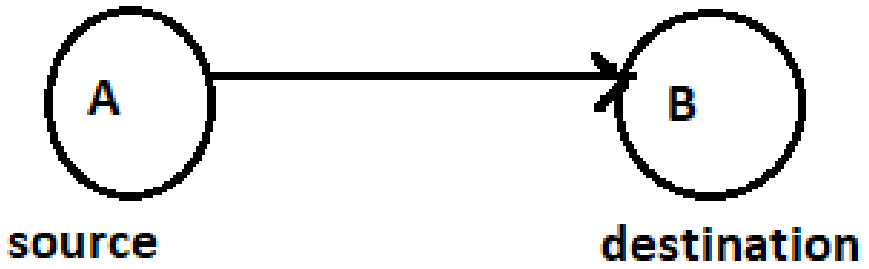
ABCD represents a path from A to D

- **Outgoing Edge**

A directed edge is said to be outgoing edge on its origin vertex.

- **Incoming Edge**

A directed edge is said to be incoming edge on its destination vertex.



- **Degree**

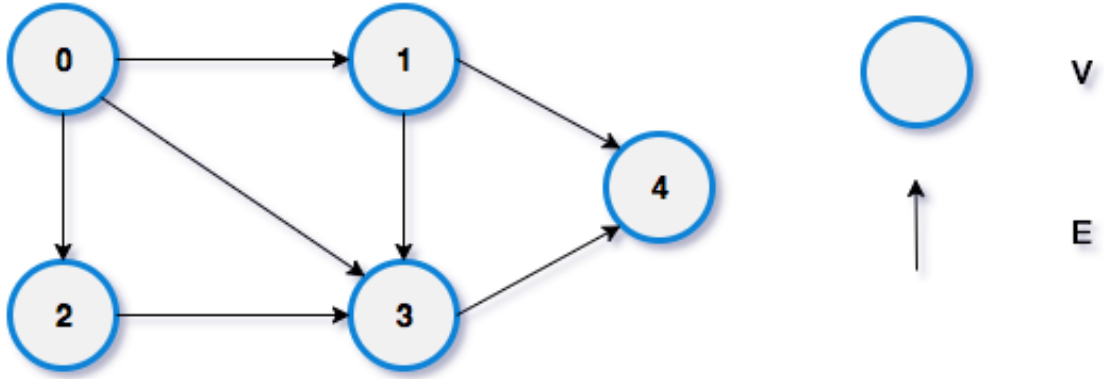
Total number of edges connected to a vertex is said to be degree of that vertex.

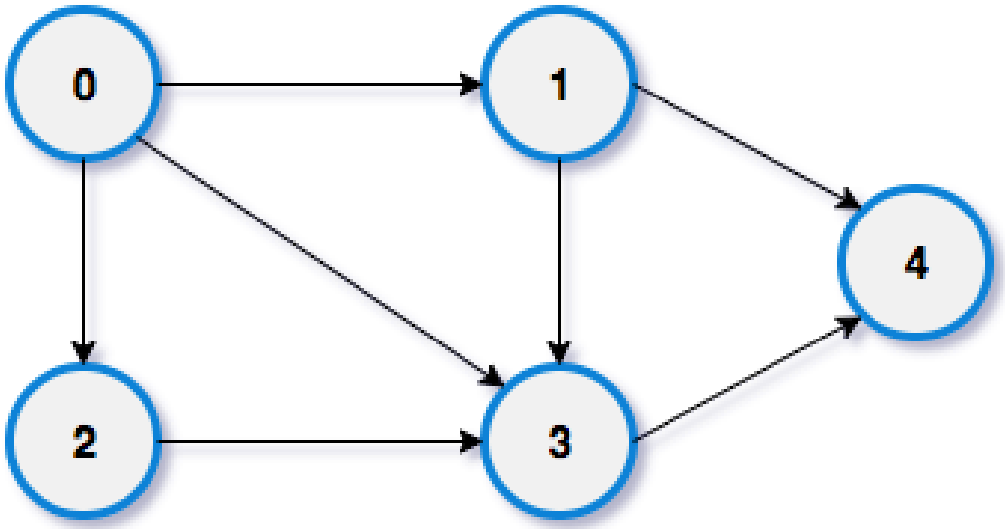
- **Indegree**

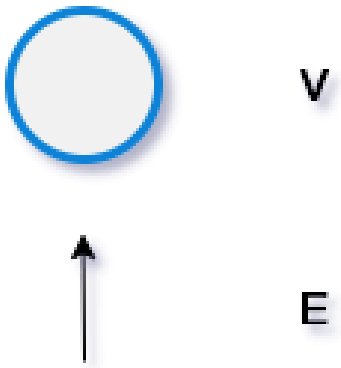
Total number of incoming edges connected to a vertex is said to be indegree of that vertex.

- **Outdegree**

Total number of outgoing edges connected to a vertex is said to be outdegree of that vertex.







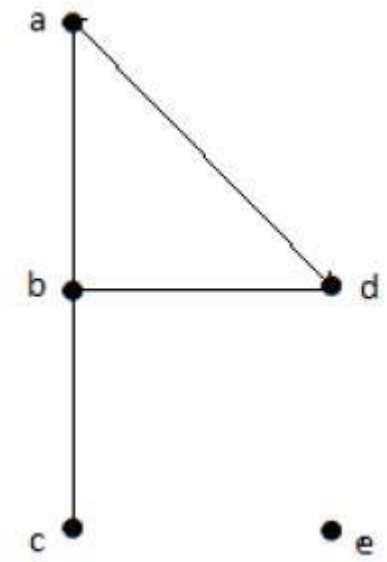
In-degree of vertex 0 = 0
In-degree of vertex 1 = 1
In-degree of vertex 2 = 1
In-degree of vertex 3 = 3
In-degree of vertex 4 = 2

Out-degree of vertex 0 = 3
Out-degree of vertex 1 = 2
Out-degree of vertex 2 = 1
Out-degree of vertex 3 = 1
Out-degree of vertex 4 = 0

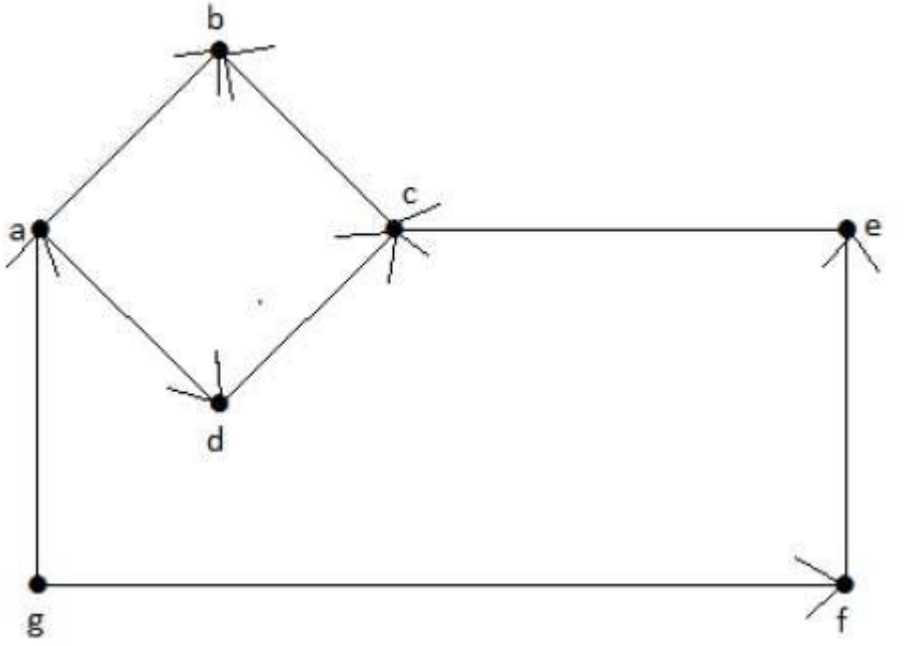
Degree of Vertex:

Degree of vertex can be considered under two cases of graphs −

- Undirected Graph
- Directed Graph



deg(a) = 2, as there are 2 edges meeting at vertex ‘a’.

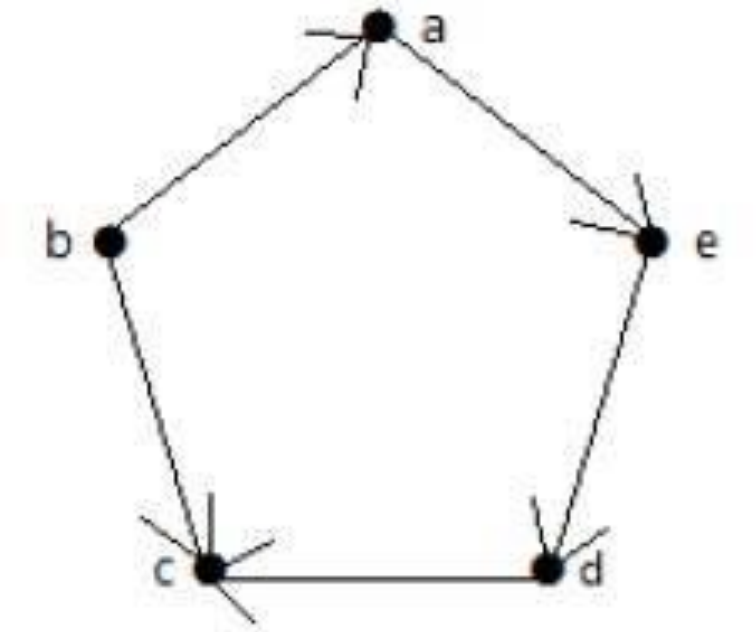


Vertex ‘a’ has two edges, ‘ad’ and ‘ab’, which are going outwards. Hence its outdegree is 2. Similarly, there is an edge ‘ga’, coming towards vertex ‘a’. Hence the indegree of ‘a’ is 1.

Real life applications use graphs:

- Graphs are nothing but connected nodes(vertex). So any network related, routing, finding relation, path etc related real life applications use graphs.
- Connecting with friends on social media, where each user is a vertex, and when users connect they create an edge.
- Using GPS/Google Maps/Yahoo Maps, to find a route based on shortest route.
- Google, to search for webpages, where pages on the internet are linked to each other by hyperlinks; each page is a vertex and the link between two pages is an edge.

Vertex	Indegree	Outdegree
a	1	2
b	2	0
c	2	1
d	1	1
e	1	1
f	1	1
g	0	2



Total indegree and outdegree of vertex e is ?

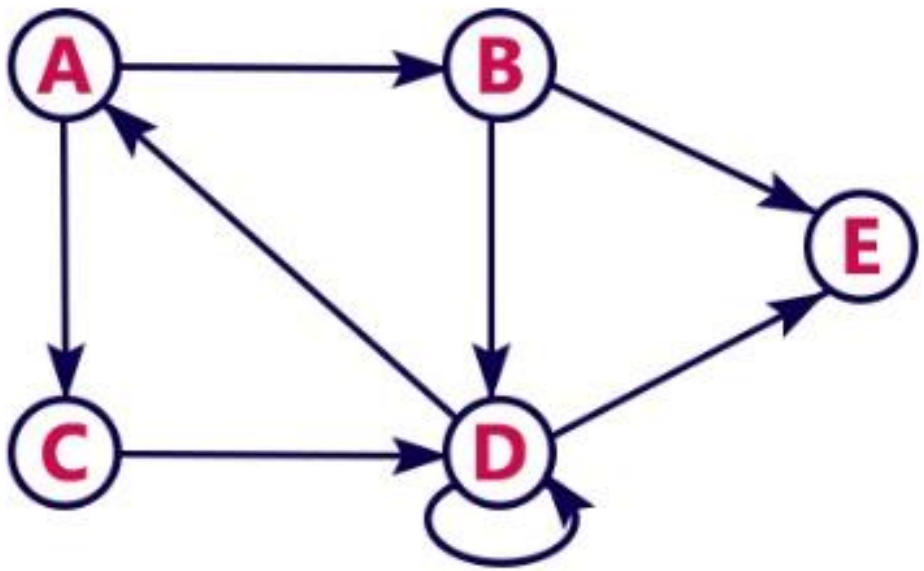
A. in-degree:0 out-degree:1
B. in-degree:1 out-degree:0
C. in-degree:0 out-degree:0
D. in-degree:1 out-degree:1

Vertex	Indegree	Outdegree
a	1	1
b	0	2
c	2	0
d	1	1
e	1	1

Graph Representations

Graph data structure is represented using following representations...

- **Adjacency Matrix**
- **Adjacency List**



	A	B	C	D	E
A	0	1	1	0	0
B	0	0	0	1	1
C	0	0	0	1	0
D	1	0	0	1	1
E	0	0	0	0	0

- **Adjacency List**
  In this representation, every vertex of graph contains list of its adjacent vertices.