Unit-6

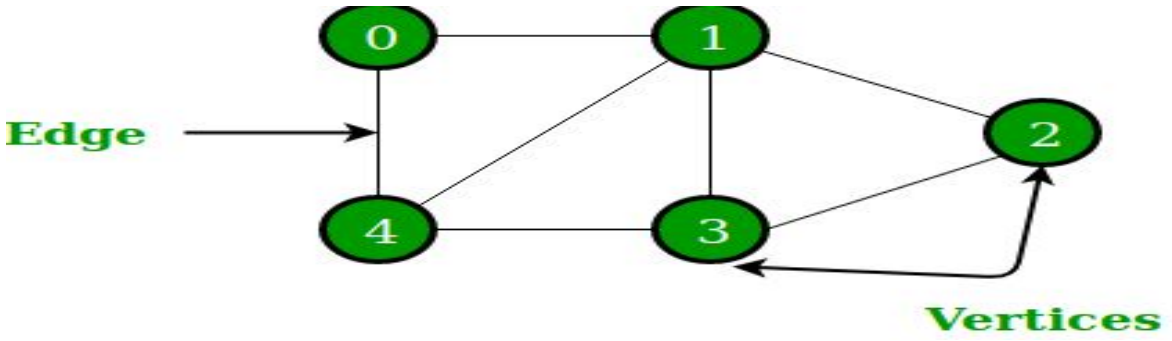
Graphs

Introduction to Graph

- A graph data structure consists of information stored in a collection of interconnected nodes(vertices) and edges(paths).
- Graph is a non-linear data structure. It contains a set of points known as nodes (or vertices) and a set of links known as edges (or Arcs). Here edges are used to connect the vertices. A graph is defined as follows...
- Graph is a collection of vertices and arcs in which vertices are connected with arcs
- Graph is a collection of nodes and edges in which nodes are connected with edges

Example

In the above Graph, the set of vertices V = {0,1,2,3,4} and the set of edges E = {01, 12, 23, 34, 04, 14, 13}.

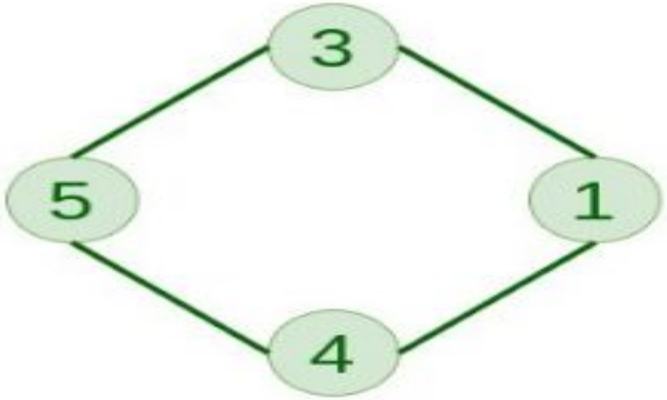


Applications of Graphs

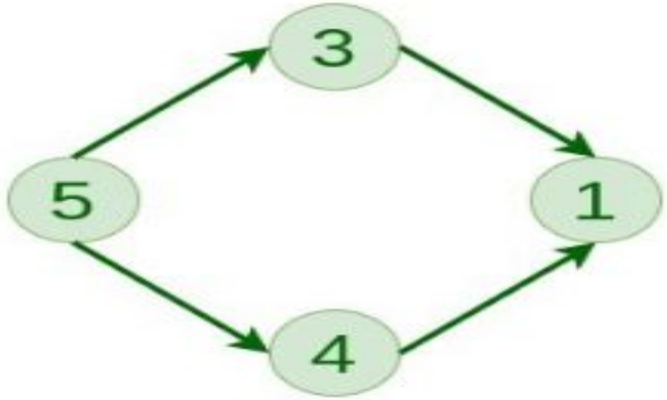
- Graphs are used to represent networks. The networks may include paths in a city or telephone network or circuit network.
- Graphs are also used in social networks like LinkedIn, Facebook. For example, in Facebook, each person is represented with a vertex(or node). Each node is a structure and contains information like person id, name, gender, and locale.
- Study molecules in chemistry & physics.
- Weighted graph used in GPS, Maps & calculate shortest path.

Types of Graph

- **Undirected Graph**
- A graph in which edges do not have any direction. That is the nodes are unordered pairs in the definition of every edge.
- **Directed Graph**
- A graph in which edge has direction. That is the nodes are ordered pairs in the definition of eve



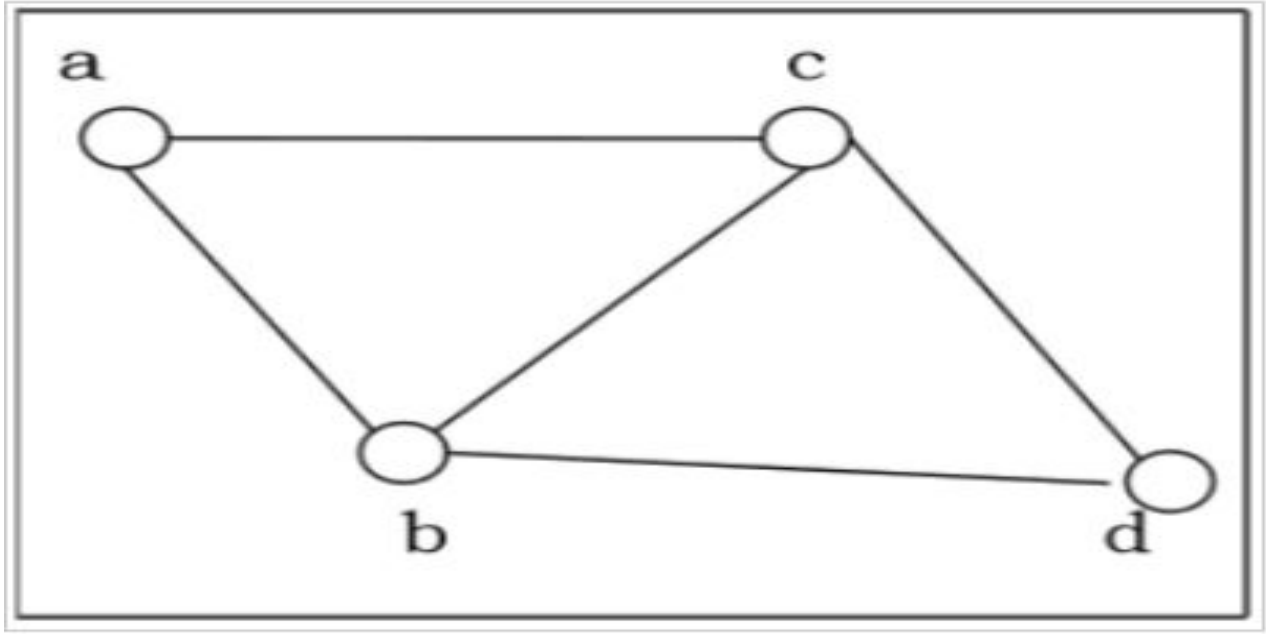
Undirected Graph



Directed Graph

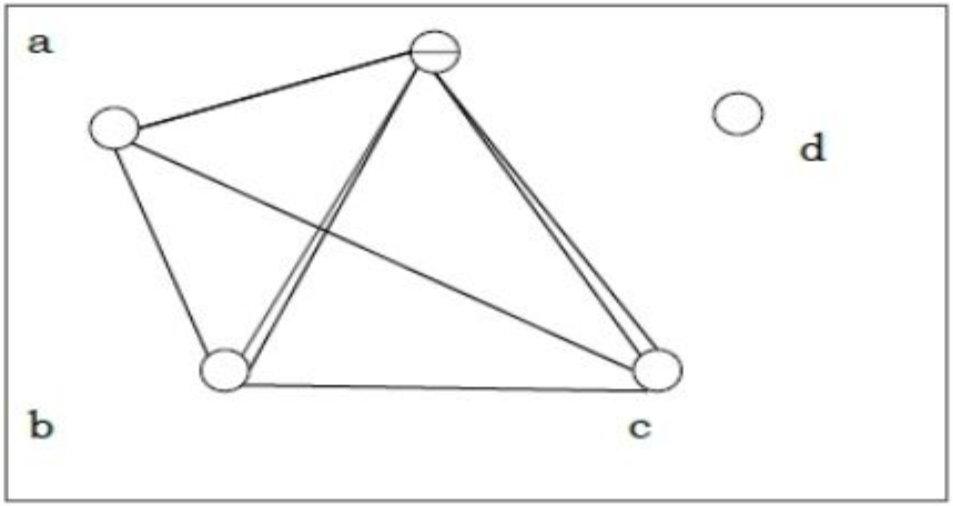
Planar graph

- A graph *G* is called a planar graph if it can be drawn in a plane without any edges crossed. If we draw graph in the plane without edge crossing, it is called embedding the graph in the plane.



Non-planar graph

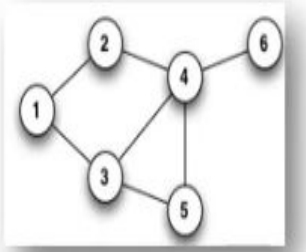
- A graph is non-planar if it cannot be drawn in a plane without graph edges crossing.



Types of Graph

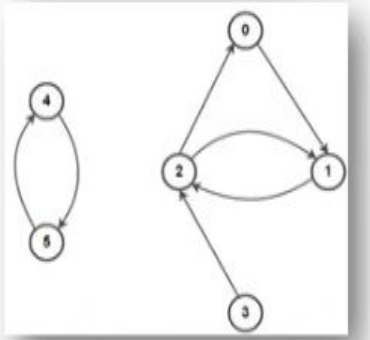
1. Undirected Graph:

Edges has no orientation.



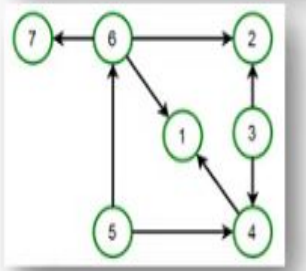
2. Directed Graph:

Edges has orientation.

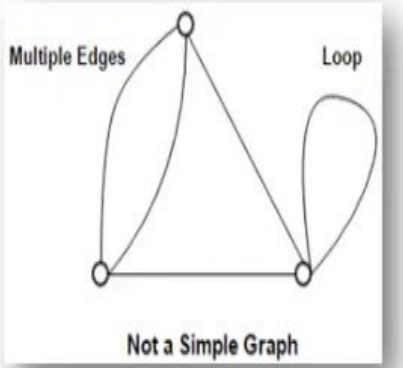


3. Directed Acyclic Graph (DAG):

No Cycle.



4. Multi Graph:



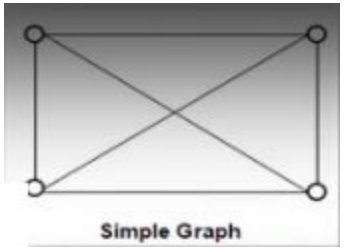
Undirected Graph, Two or more edges connected to the same vertices, Loops allow.

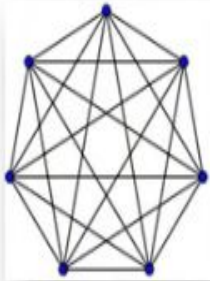
Types of Graph

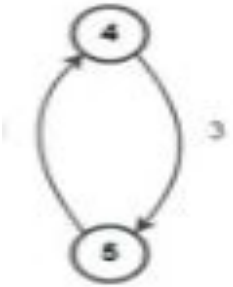
5. Simple Graph: Undirected graph. Loops are not allow.

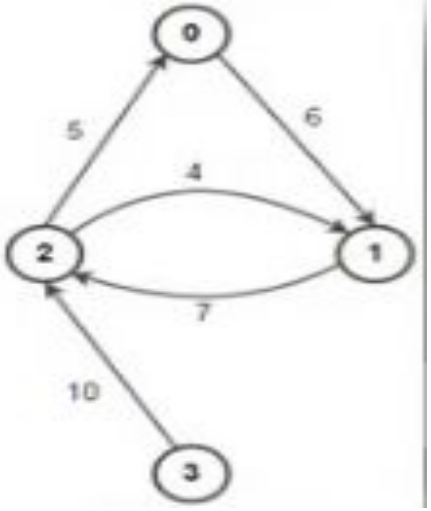
6. Weighted Graph: Weight or value assign to each edge.

7. Complete Graph: All edges are connected to each vertices.



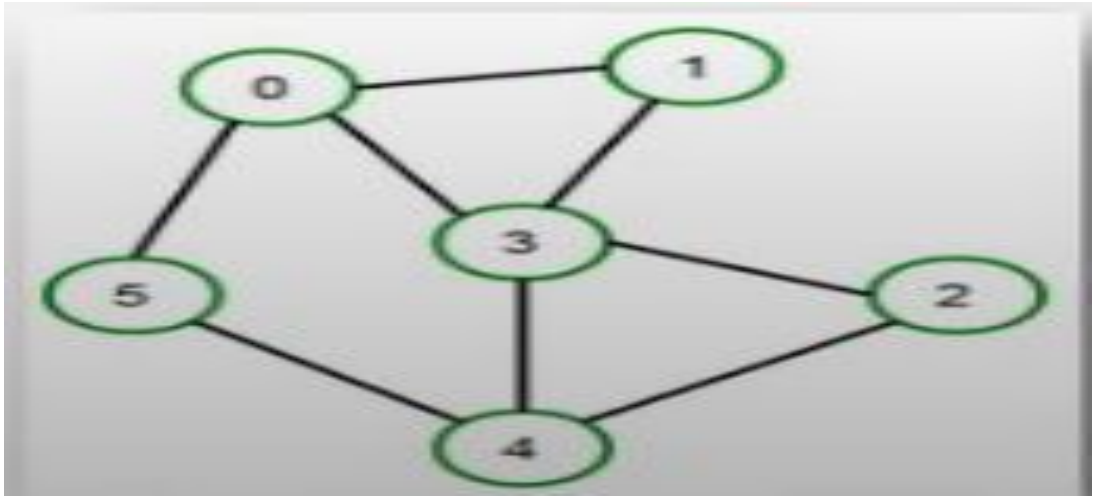






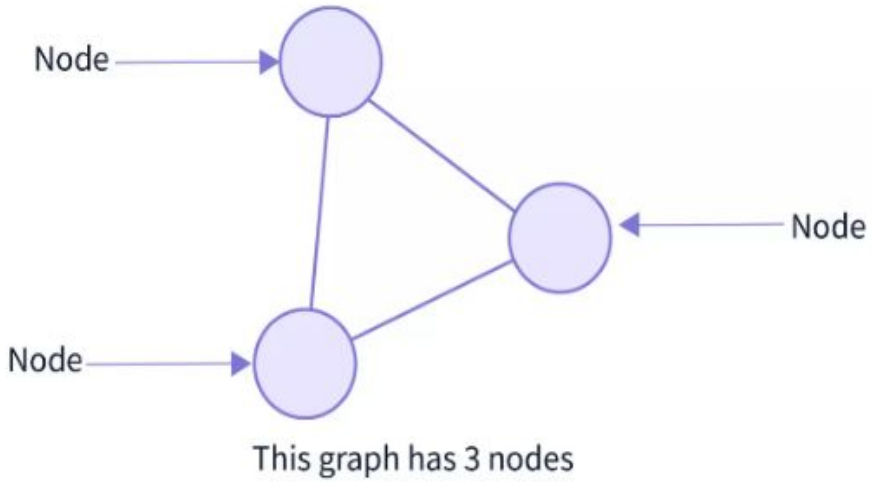
Types of graph

8. Connected Graph: Path between every pair of vertices.



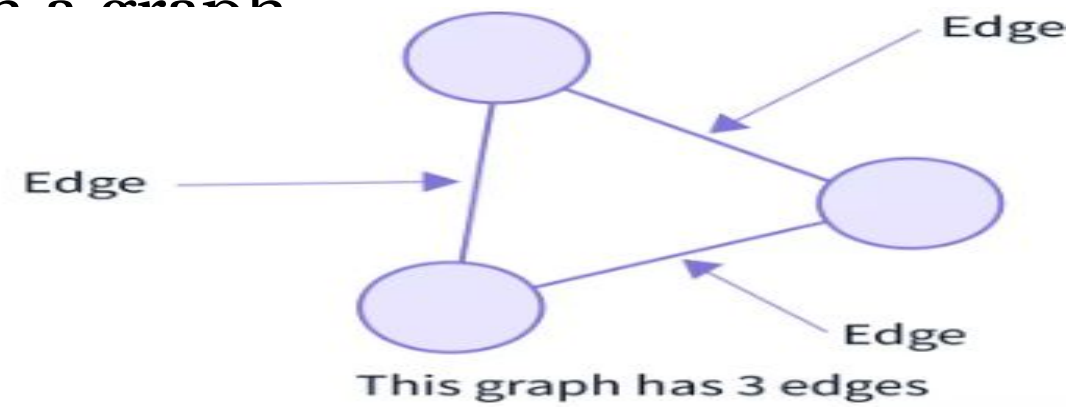
Nodes:

- Nodes create complete network in any graph. They are one of the building blocks of a graph data structure. They connect the edges and create the main network of a graph. They are also called **vertices**.
- A node can represent anything such as any location, port, houses, buildings, landmarks, etc. They basically are anything that you can represent to be connect
and you can establish a
them.



Edges:

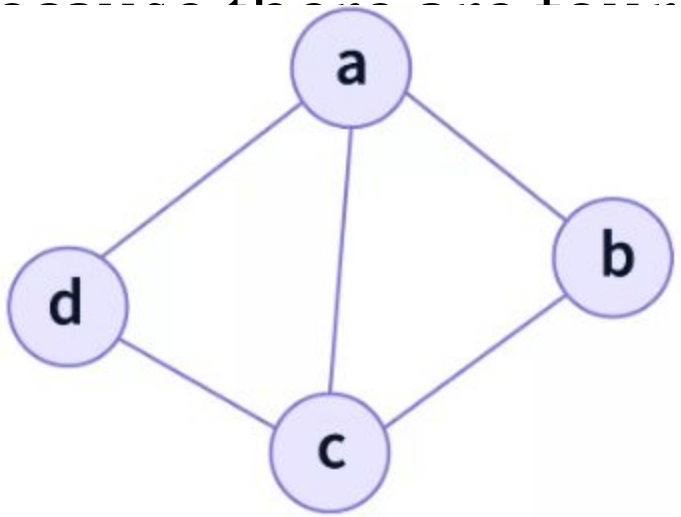
- Edges basically connects the nodes in a graph data structure. They represent the relationships between various nodes in a graph. Edges are also called the path in a graph



- The above image represents edges in a graph.
- A graph data structure (V, E)(V,E) consists of:
- A collection of vertices (V)(V) or nodes.
- A collection of edges (E)(E) or path

Example:

- The below image represents a set of edges and vertices:
- A graph is a pair of sets (V, E), where V is the set of **vertices** and E is the set of **edges**, connecting the pairs of vertices. In the above graph:
- V = {a, b, c, d} E = {ab, ac, ad, bc, cd}
- In the above graph, |V| = 4 b[illegible] nodes (vertices) and, |E| = 5 b[illegible] edges (lines).



Graph Terminologies

1. **Node:** Every individual elements or vertex in graph. Ex. A,B,C,D

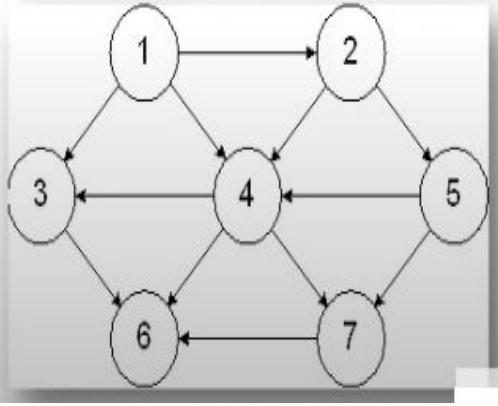
2. **Arc(Edges):** Link between two vertices. 1->2->5

3. **Directed Edge:** Gives specific direction.

4. **Undirected Edge:** Does not show any direction.

5. **Degree:** Total no. of edges connected to the vertex.

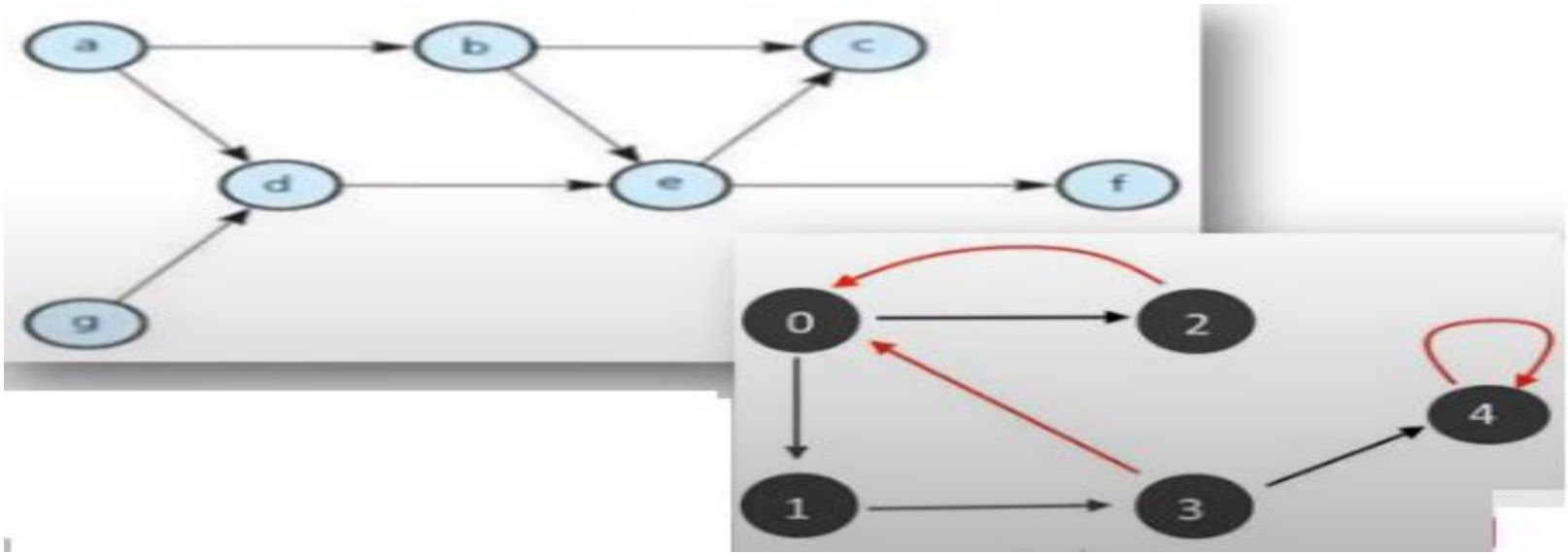
6. **In-Degree:** Total No. of incoming edges connected to vertex.



7.**Out-Degree**:Total No.of outgoing edges connected to vertex

Graph Terminology

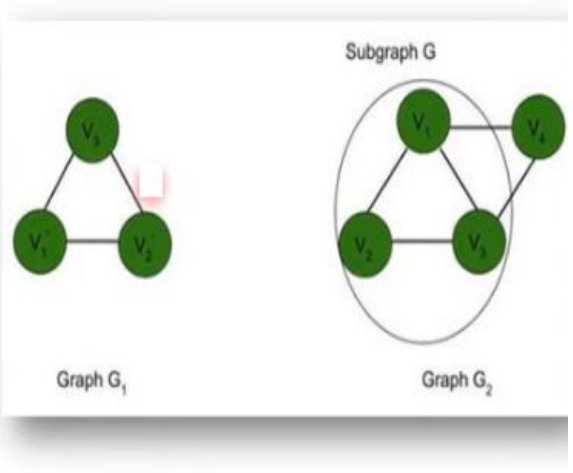
13. Linear Path: The path which starting & ending vertex is different.



14.Cyclic :The path which starting and ending vertex is same

Graph Terminology

- **15. Sub Graph:** Subsets of graph.

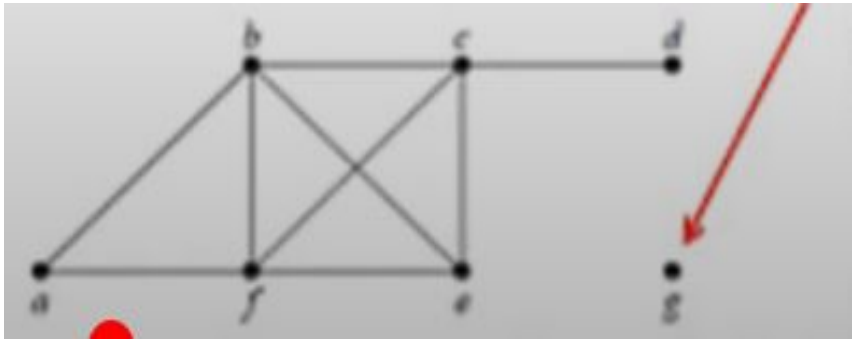


16. Source: Vertex with no incoming edges. In degree is 0.

17. Isolated Node: Single node. Vertex having degree as zero.

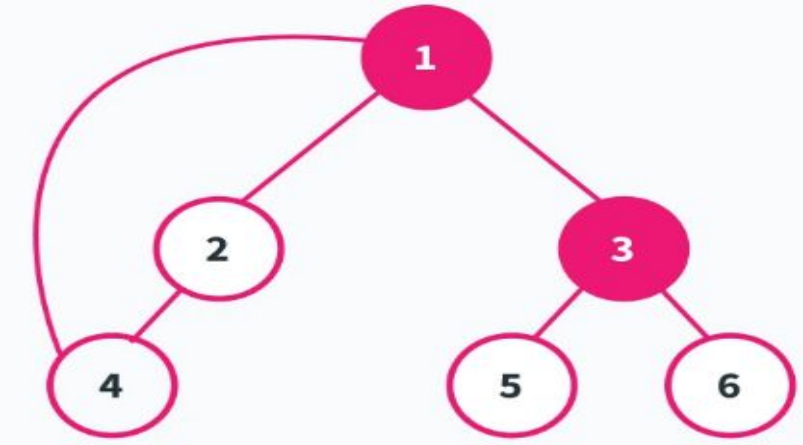
18. Sink: In directed graph, Vertex has only incoming edge not outgoing.

19. Articulation point: Vertex in connected graph is remove then disconnect the graph.

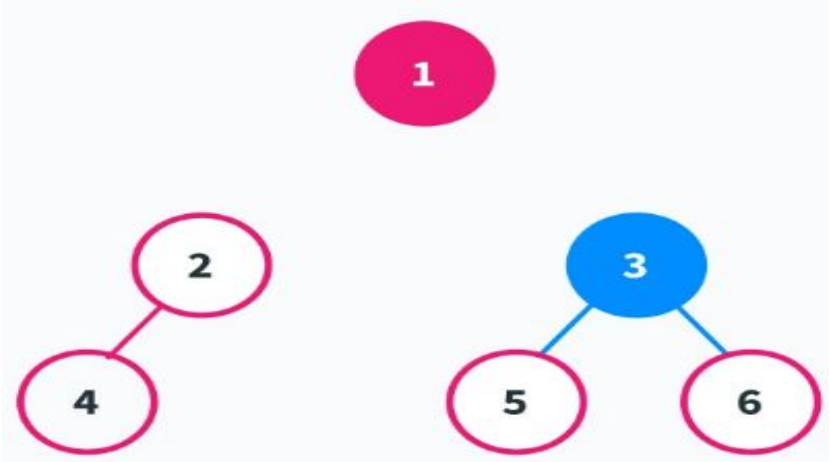


Graph Terminology

- Articulation point:We can see in this graph that the number of connected components is 1.



- Removing vertex 1 from gra



- In this graph, we can see that removing vertex 1 results in the generation of two connected components **2---4** and **5---3---6**. Thus vertex 1 is an articulation point

Graph Representations

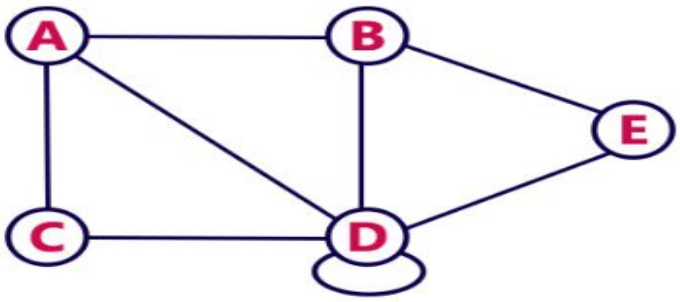
Graph data structure is represented using following representations...

- Adjacency Matrix
- Incidence Matrix
- Adjacency List

Graph Representations

- **Adjacency Matrix: In** this representation, the graph is represented using a matrix of size total number of vertices by a total number of vertices. That means a graph with 4 vertices is represented using a matrix of size 4X4.
- In this matrix, both rows and columns represent vertices. This matrix is filled with either 1 or 0. Here, 1 represents that there is an edge from row vertex to column vertex and 0 represents that there is no edge from row vertex to column vertex.
- I

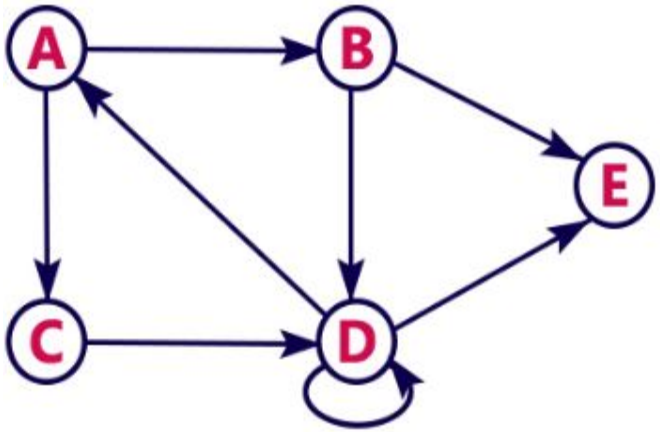
r



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

Graph Representations

- Directed graph representation.

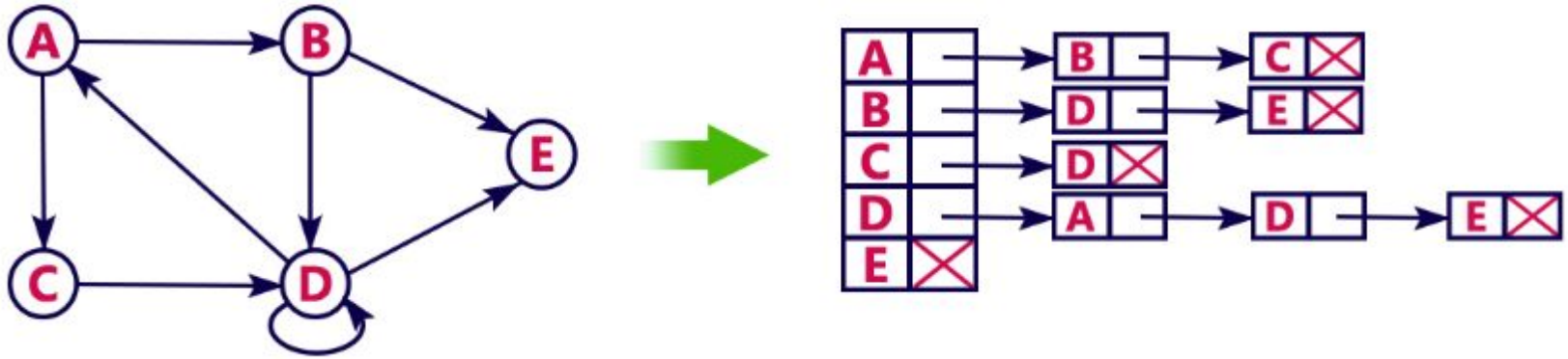


	A	B	C	D	E
A	0	1	1	0	0
B	0	0	0	1	1
C	0	0	0	1	0
D	1	0	0	1	1
E	0	0	0	0	0

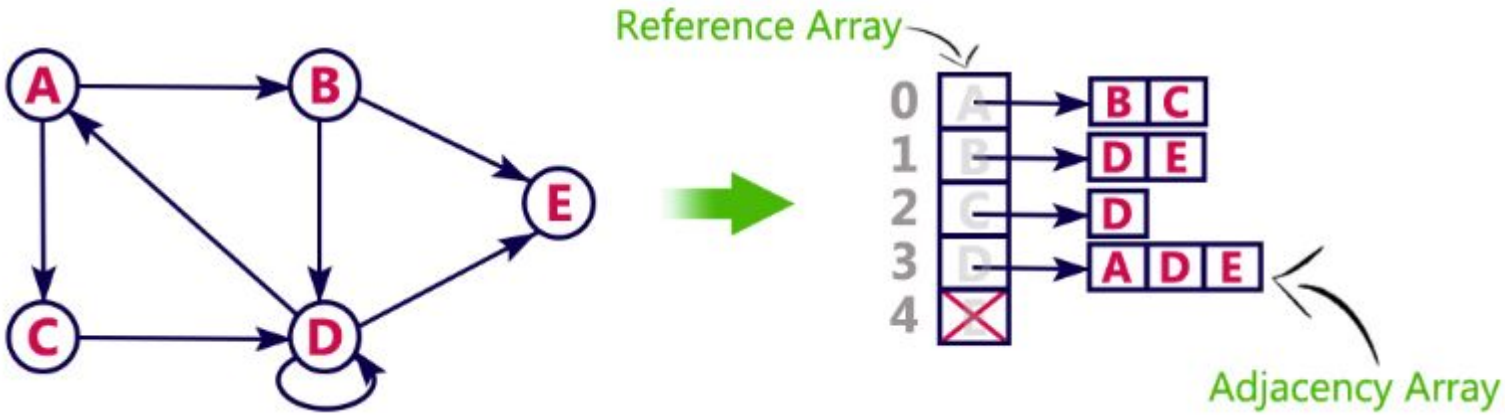
Adjacency List

- In this representation, every vertex of a graph contains list of its adjacent vertices.

For example, consider the following directed graph representation implemented using linked list...

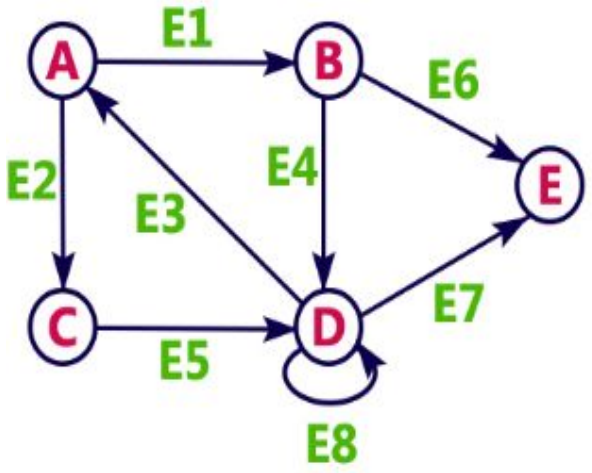


This representation can also be implemented using an array as follows..



Incidence Matrix

- In this representation, the graph is represented using a matrix of size total number of vertices by a total number of edges. That means graph with 4 vertices and 6 edges is represented using a matrix of size 4X6. In this matrix, rows represent vertices and columns represents edges. This matrix is filled with 0 or 1 or -1. Here, 0 represents that the row edge is not connected to column vertex, 1 represents that the row edge is connected as the outgoing edge to column vertex and -1 represents that the row edge is connected as the incoming



	E1	E2	E3	E4	E5	E6	E7	E8
A	1	1	-1	0	0	0	0	0
B	-1	0	0	1	0	1	0	0
C	0	-1	0	0	1	0	0	0
D	0	0	1	-1	-1	0	1	1
E	0	0	0	0	0	-1	-1	0