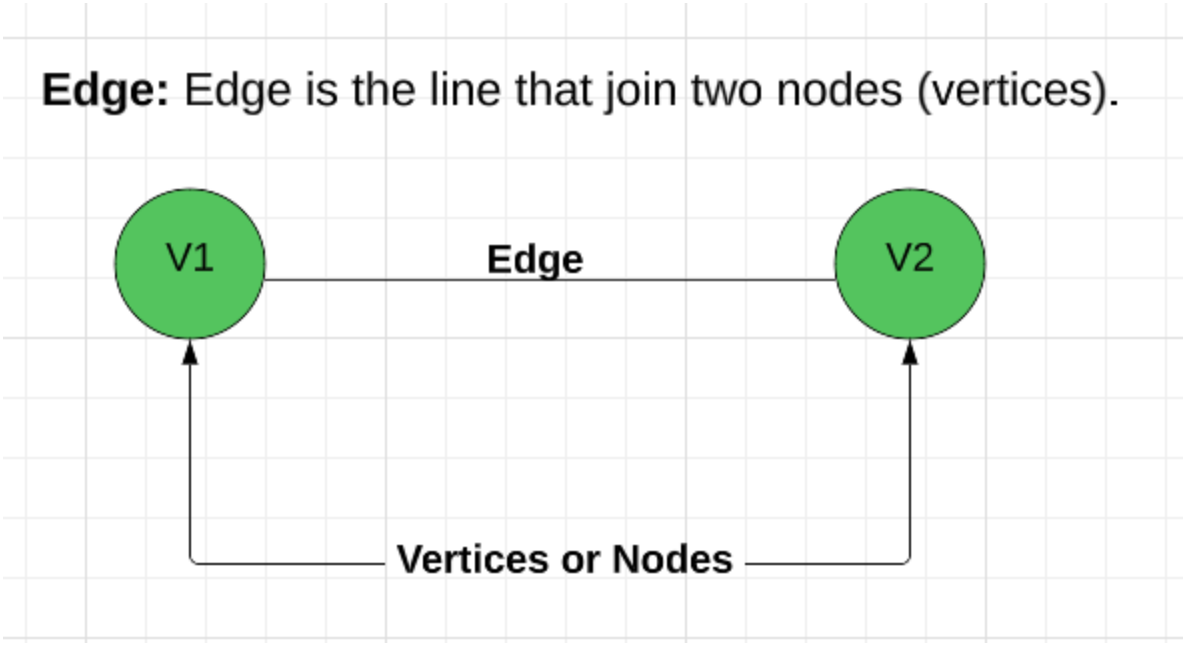
Graph Data Structure

What is Graph ?

“A graph is a non-linear data structure that consists of a set of nodes (also called vertices) and a set of edges that connect pairs of nodes. A node can represent any kind of object or entity, and the edges represent the relationships between them”.

Important Points:

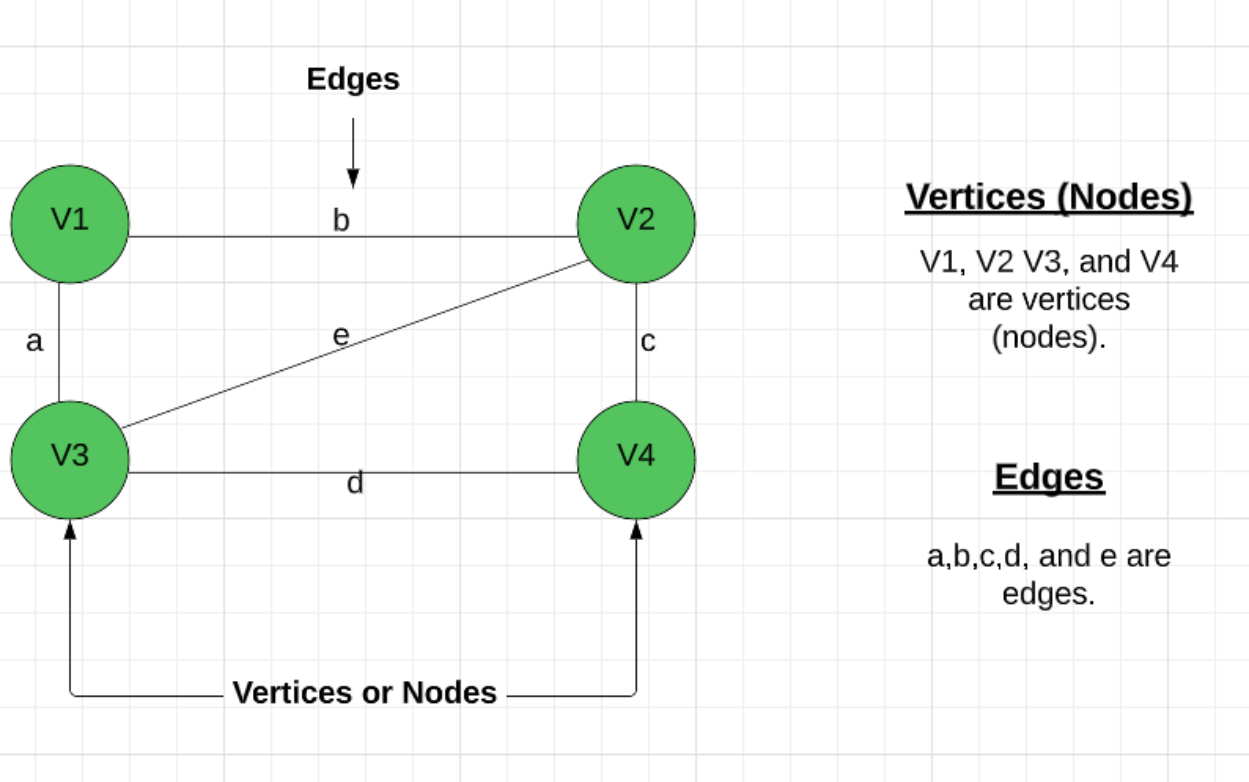
1. Nodes in Graphs are sometimes also referred to as vertices.
2. Edges are lines or arcs that connect any two nodes in the graph.



3. In Simple words, A graph is composed of a set of vertices(V) and a set of edges(E).
4. The graph is denoted by **G = (V,E)**, where G represents the graph, V represents the set of vertices (nodes), and E represents the set of edges that connect the vertices.

$$G = (V, E)$$

Example:



In the above graph,

Set of Vertices (nodes) are: V = {V1, V2, V3, V4, V4}.

Set of Edges are: E = {a, b, c, d, e}.

Graph G = (V, E).

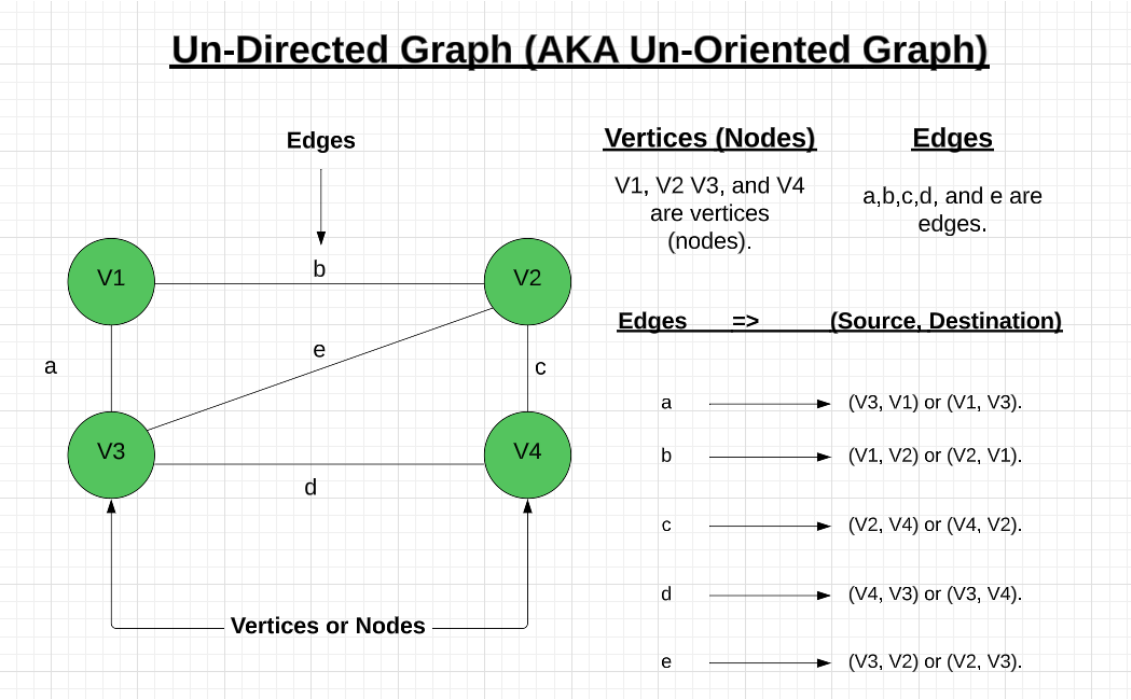
Components of a Graph:

- **Vertices:** a node (also known as a vertex or point) represents an entity or object. Nodes are connected to each other by edges, and together they form the structure of the graph. Each node can have its own properties, such as a label or value, that help to distinguish it from other nodes in the graph.
- **Edges:** Edges are used to connect two nodes in the graph. The edges can be directed or undirected, and can have a weight or not.

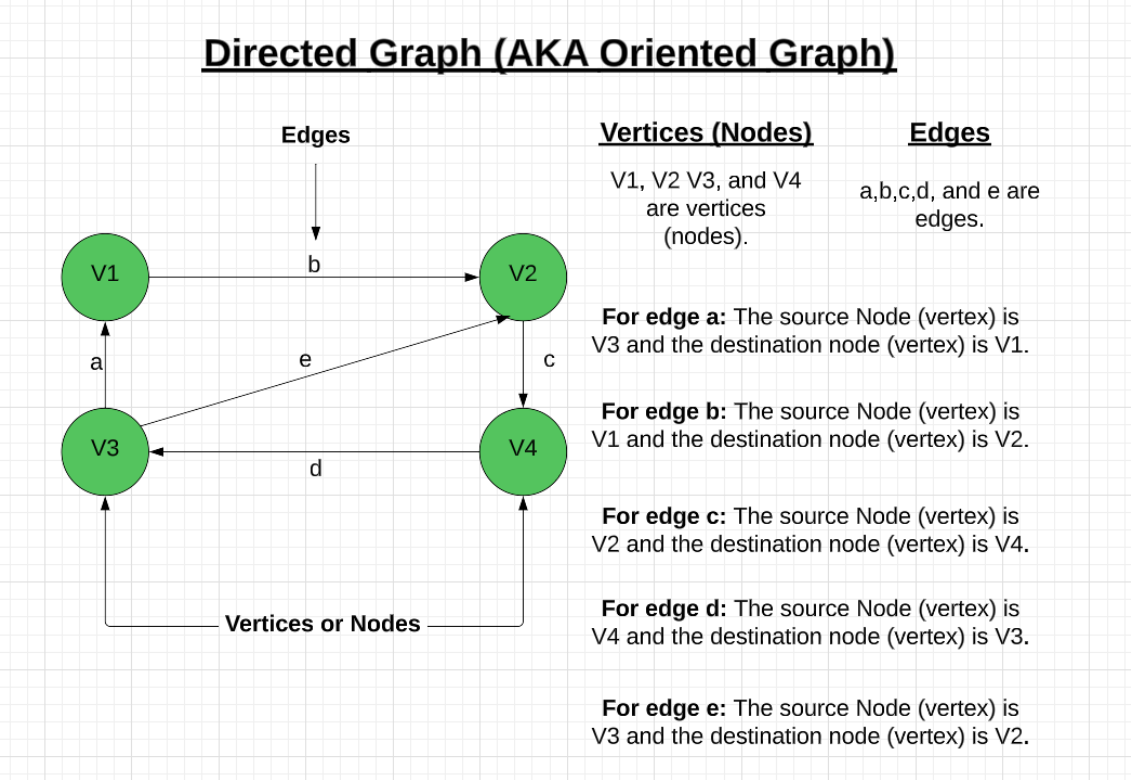
Types of Graphs:

There are several types of graphs, each with its own characteristics and properties. Some of the most common types of graphs are:

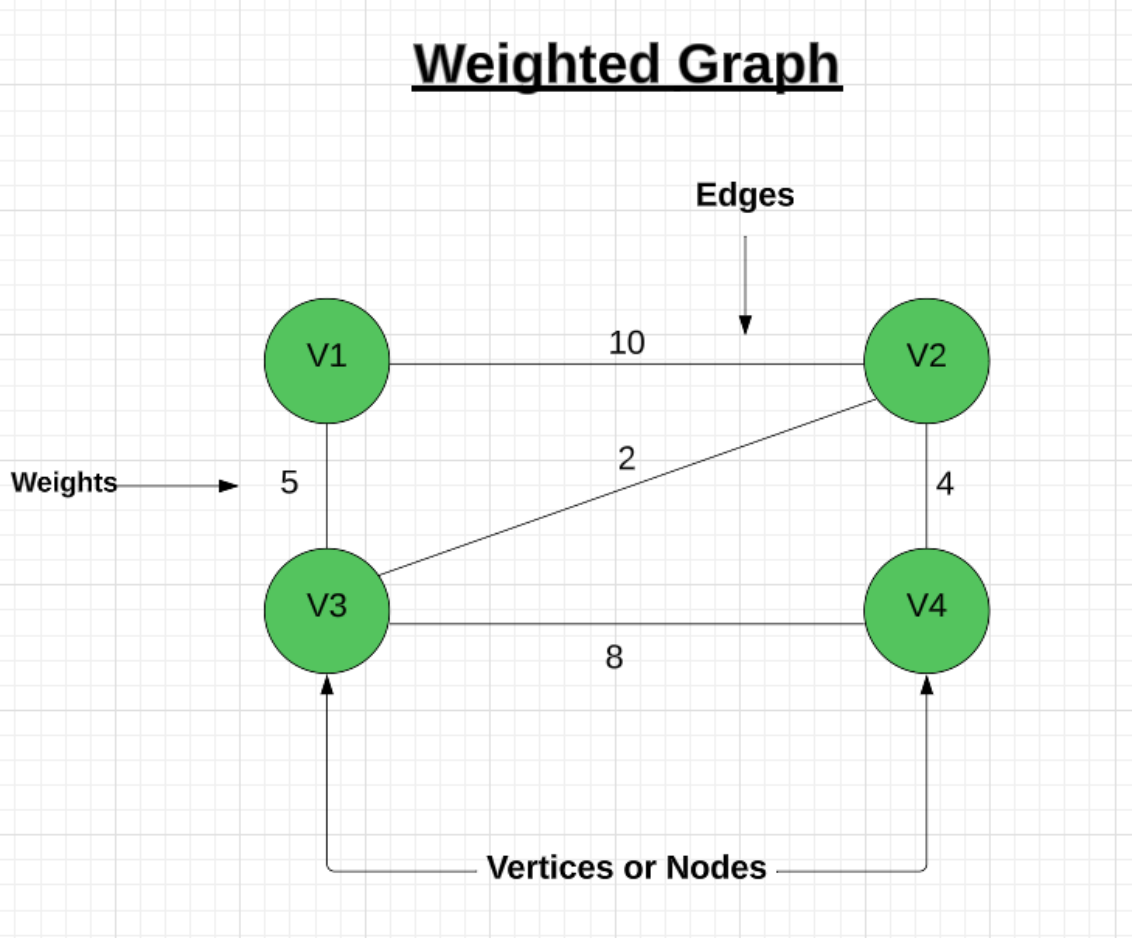
1. **Undirected Graph:**
    a. In an undirected graph, the edges have **no direction.** This means that the connection between two nodes is **bidirectional**, and there is no concept of a source or a destination node.



2. **Directed Graph:**
    a. In a directed graph, the edges have a **direction**. This means that the connection between two nodes is **unidirectional**, and there is a concept of a source and a destination node.



3. **Weighted Graphs:**
    a. A graph in which edges have weights or costs associated with them. Example: A road network graph where the weights can represent the distance between two cities.



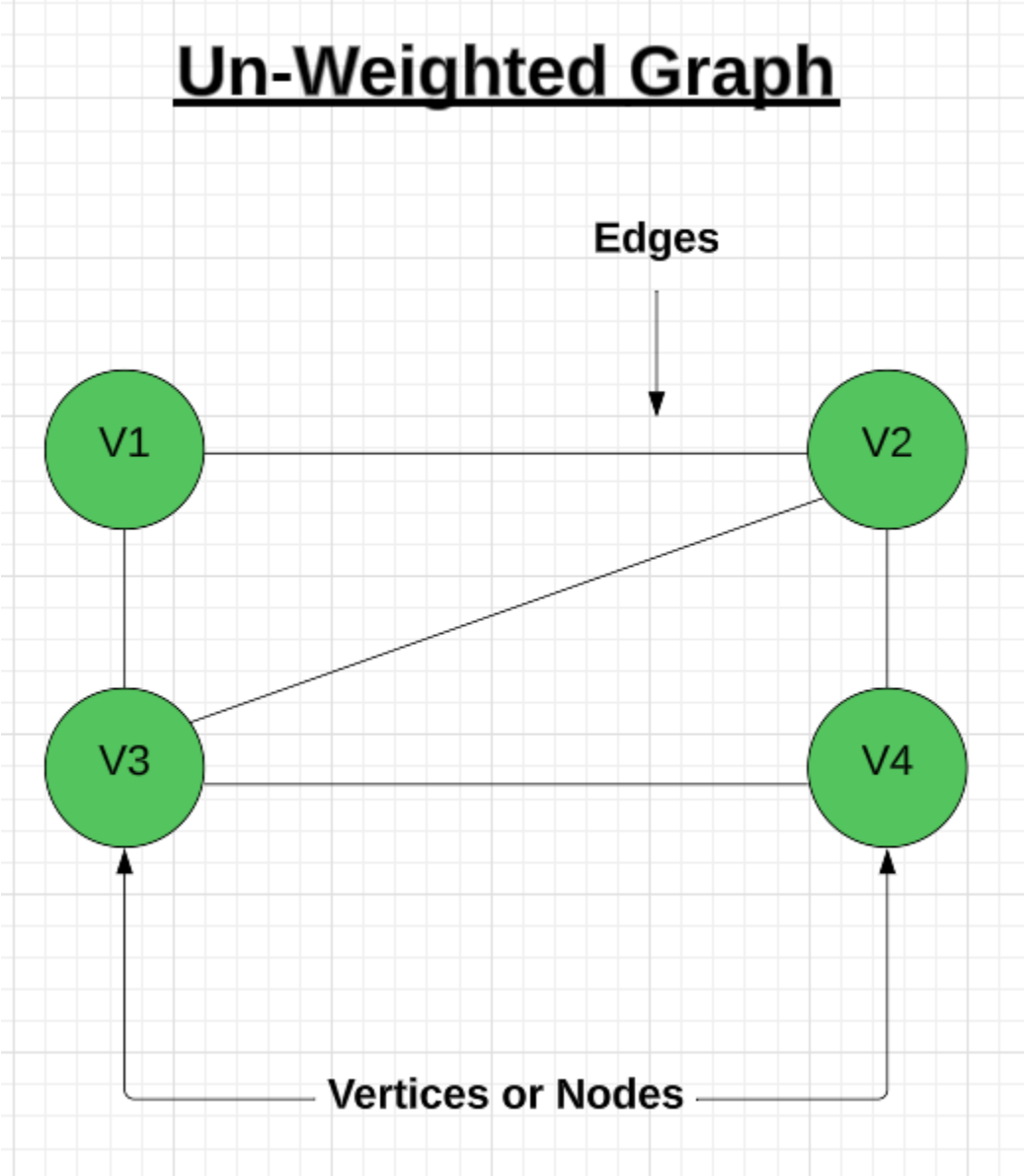
b. You can further divide this weighted graph into Weighted Directed Graph and Weighted Undirected Graph and the only difference between both the graphs is direction in edges.

c. In a weighted directed graph, each edge has a direction and a weight associated with it.

d. In an undirected weighted graph, each edge has a weight associated with it, but there is no direction associated with the edges.

4. **Unweighted Graphs:**

a. A graph in which edges have no weights or costs associated with them. Example: A social network graph where the edges represent friendships.



b. You can further divide this weighted graph into Un-Weighted Directed Graph and Un-Weighted Undirected Graph and the only difference between both the graphs is direction in edges.

c. In an unweighted directed graph, each edge has a direction, but no weights associated with it.

d. In an unweighted undirected graph, each edge has no direction or weight associated with it.

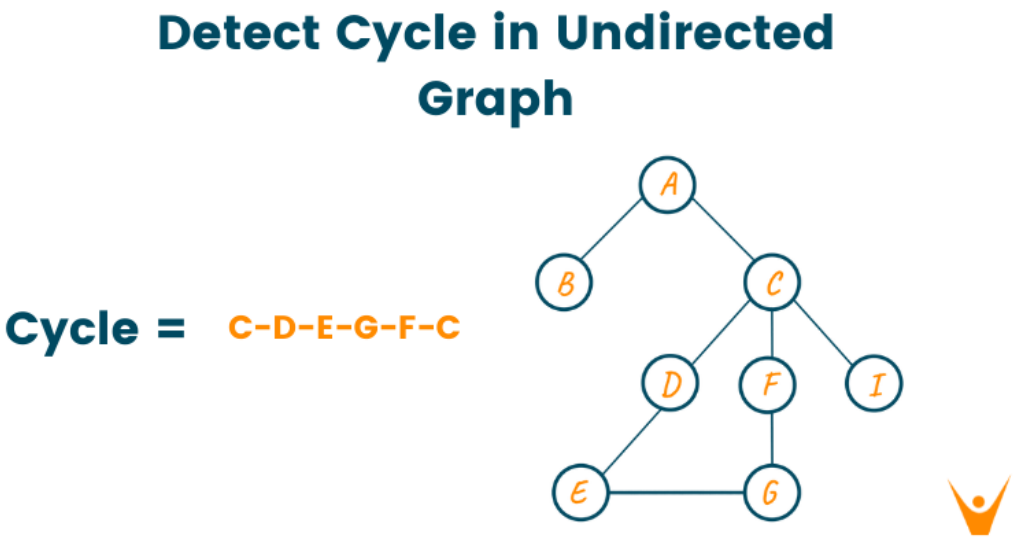
5. **Cyclic Graph:**

a. A cyclic graph is a graph that has at least one cycle. A cycle is a path that starts and ends at the same node, and no other node appears twice in the path.

b. A cyclic graph can be either directed or undirected.

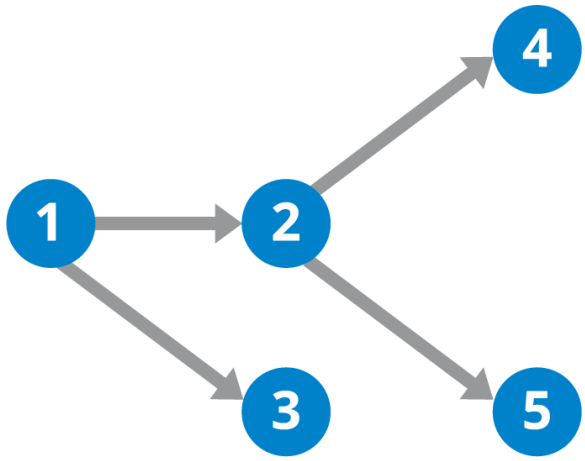
   i. In a directed cyclic graph, the edges have a direction, and the cycle must follow the direction of the edges.

   ii. In an undirected cyclic graph, the edges have no direction, and the cycle can go in any direction.



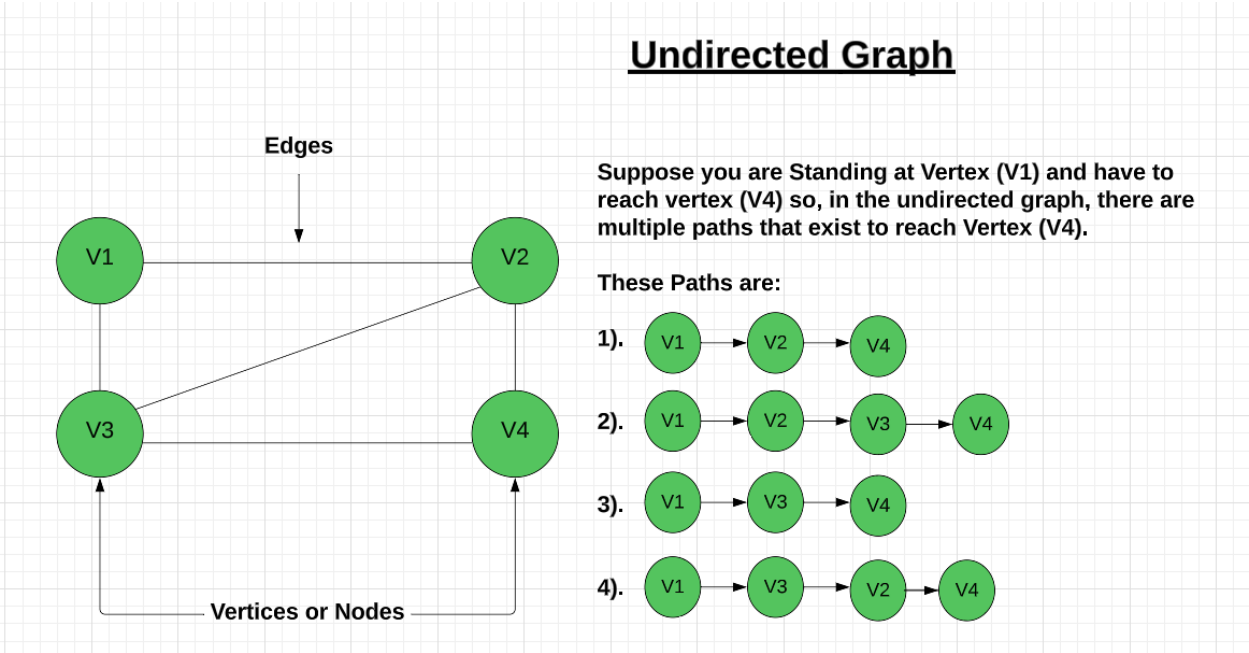
6. **Acyclic Graph:**

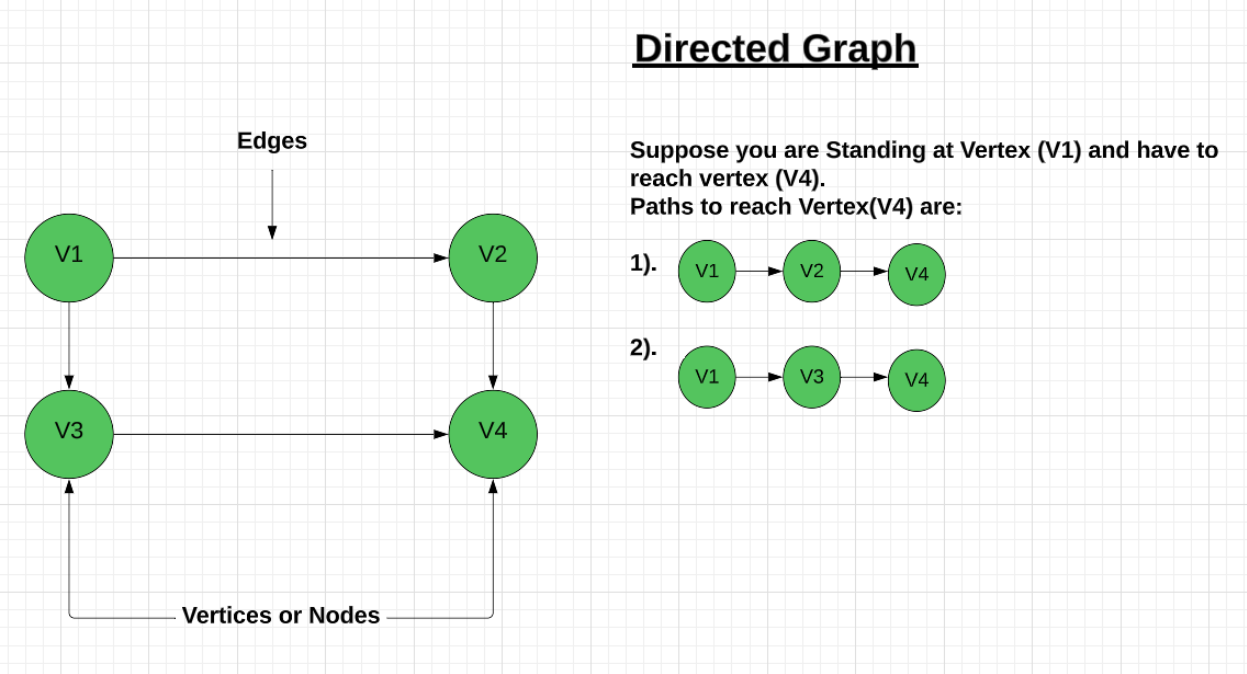
   a. In an acyclic graph, there are no cycles. This means that there is no path that starts and ends at the same node.



What is Path?

In graph theory, a path is a sequence of connected edges in a graph that allows you to travel from one vertex (or node) to another vertex in the graph. A path may visit the same vertex multiple times, **but it cannot repeat the same edge more than once.**

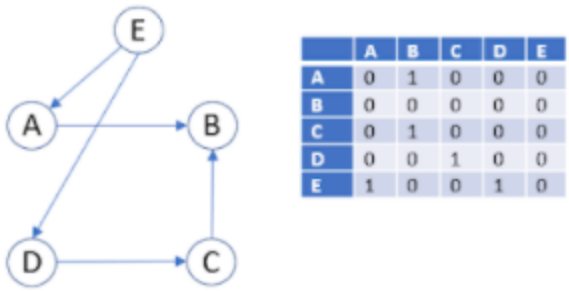




Graph Representation:

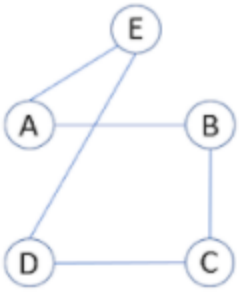
Graphs can be represented in various ways, including:

1. **Adjacency Matrix**
    a. An adjacency matrix is a square matrix that represents a graph by using a 2D array. The rows and columns of the matrix represent the vertices of the graph, and each element (i, j) of the matrix indicates whether there is an edge between vertices i and j. If there is an edge between vertices i and j, the element (i, j) is set to 1, otherwise it is set to 0.



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

Fig 3: Adjacency Matrix for a directed graph



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

Fig 4: Adjacency Matrix for an undirected graph

2. **Adjacency List**

   a. An adjacency list is a collection of linked lists, where each linked list represents the neighbors of a vertex in the graph. Each vertex in the graph is associated with a linked list that contains the vertices it is adjacent to.

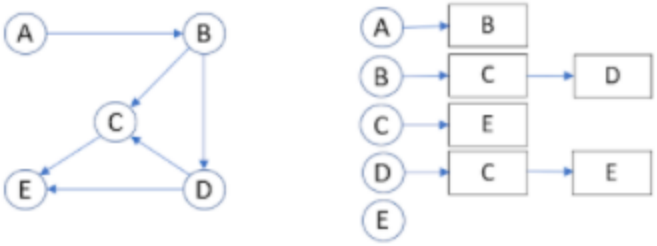


Fig 6: Adjacency list for a directed graph

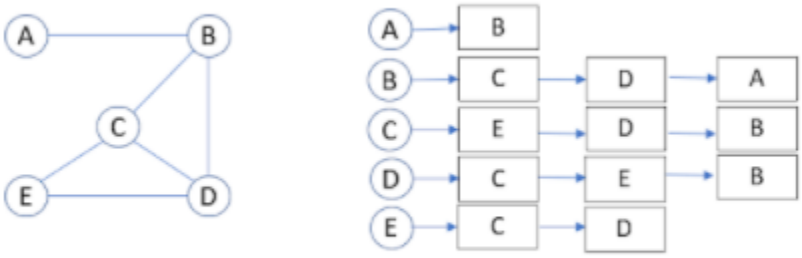


Fig 7: Adjacency list for an undirected graph