By Vishal Sharma

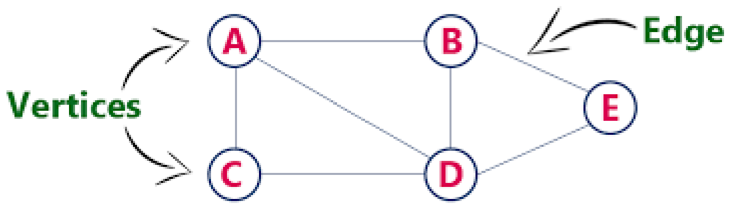
Graph Data Structure

Graph is basically a network of nodes in which various nodes are interconnected with each other.

Each node in graph is known as vertex, and the connection between each vertex is known as edge.

Each vertex contains a particular data in it.

Example: -

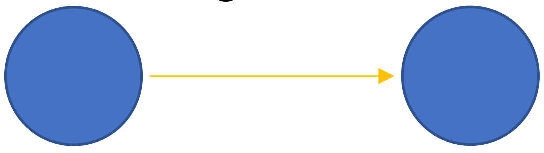


Edges: - The linking between vertices is known as edge.

Degree: - The degree of any vertex is the no. of vertices connected with it. Example: - Degree of A is 3 because 3 vertices is connected with A and that is (B, C, D).

There are two types of edges: -

1. **Uni-directional edge: -**



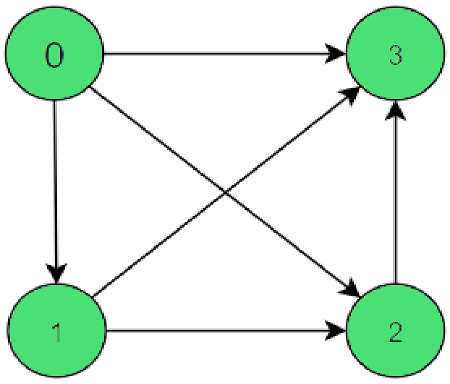
The Graph having Uni-directional edges in it is known as **Directed Graph.**

In Directed Graph Degree are of two types: -

In-Degree: - Number of edges pointing towards a particular vertex.

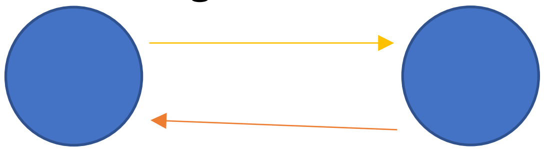
Out-Degree: - Number of edges not pointing towards a particular vertex.

Example: -



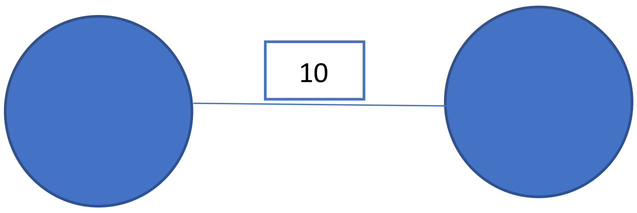
In-Degree of 0 is 0	Out-Degree of 0 is 3
In-Degree of 1 is 1	Out-Degree of 1 is 2
In-Degree of 2 is 2	Out-Degree of 2 is 1
In-Degree of 3 is 3	Out-Degree of 3 is 0

2. **Bi-directional edge or Undirectional edge: -**

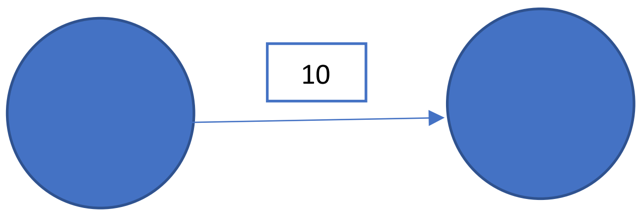


The Graph having Bi-directional edges in it is known as **Undirected Graph.**

3. Weighted Graph: - In this type of graph, Edge contain weight.

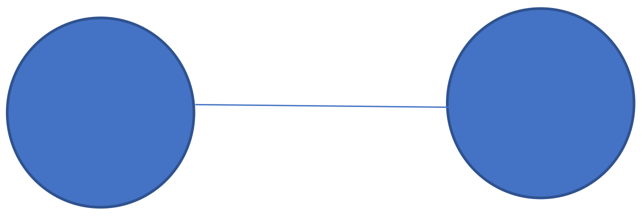


This is an example of undirected weighted graph.

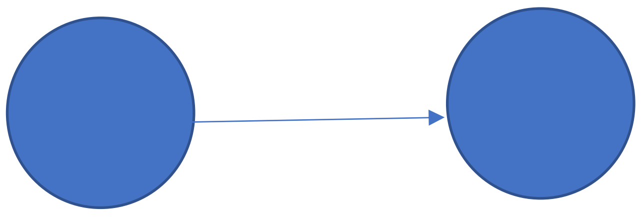


This is an example of directed weighted graph.

4. Un-weighted Graph: - In this type of graph, Edge does not contain any type of weight.



This is an example of undirected Un-weighted graph.

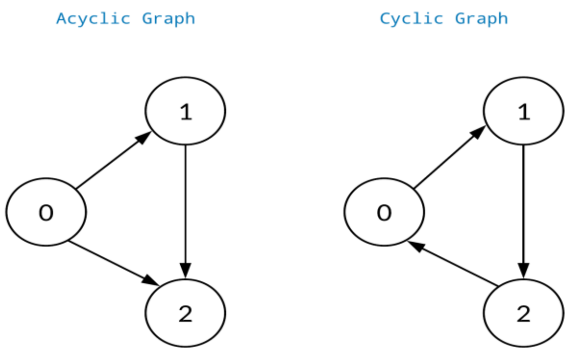


This is an example of directed Un-weighted graph.

Cyclic-Graph: - This Graph type have the source and destination position same means the starting and the ending position of the graph are same.

Acyclic-Graph: - This Graph type does not have the source and destination position same means the starting and the ending position of the graph are not same.

Example of cyclic graph and Acyclic graph: -



Real Life applications of Graph Data Structures: -

1. In Google Maps
2. Uber and Ola apps
3. Zomato and Swiggy apps
4. In Social Media like Instagram and Facebook etc.

As you know that when we create a binary tree then we create a structure or class which contains data value, left pointer and right pointer.

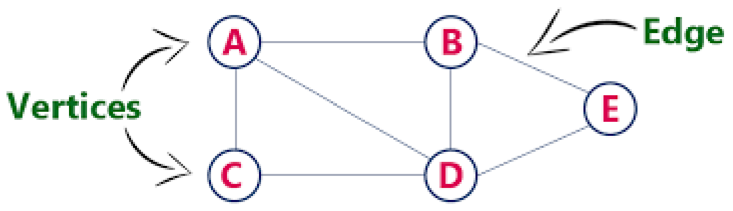
Similarly, there are Four ways for creating Graph

1. **Adjacency List (**Commonly used method for creating graph**)**
2. **Adjacency Matrix**
3. **Edge List**
4. **2D matrix (Implicit Graph).**

Adjacency List: - It is also known as List of Lists.

In this List we store the list of neighbours of vertices.

Example: -

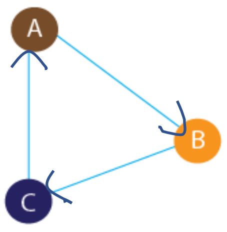


In this diagram the neighbour of A is B, C and D and a list formed according to the given data is
Vertex → {Source, Destination} A → {A, B}, {A, C}, {A, D}
Similarly, neighbours of B, C, D, E are
B → {B, A}, {B, D}, {B, E}
C → {C, A}, {C, D}
D → {D, A}, {D, B}, {D, C}, {D, E}
E → {E, B}, {E, D}

The first line of Adjacency List said that it is a list of lists
Hence the proper Adjacency form for the above example is: -
{{A, B}, {A, C}, {A, D}, {B, A}, {B, D}, {B, E}, {C, A}, {C, D}, {D, A}, {D, B}, {D, C}, {D, E}, {E, B}, {E, D}}

Adjacency Matrix: - In this implementation of Graph matrix is formed in which all the vertices and their neighbour are stored in the form of a matrix just like sparse matrix.

Example: -



This example is expressed in a matrix in the form of Adjacency matrix: -

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

1 represent in the specific row and column that the column value is the neighbour of the row. (that means B in first row is the neighbour of A)
And 0 represent that the particular column value if not the neighbour of the row. (that means A and C in first row is not the neighbour of A)