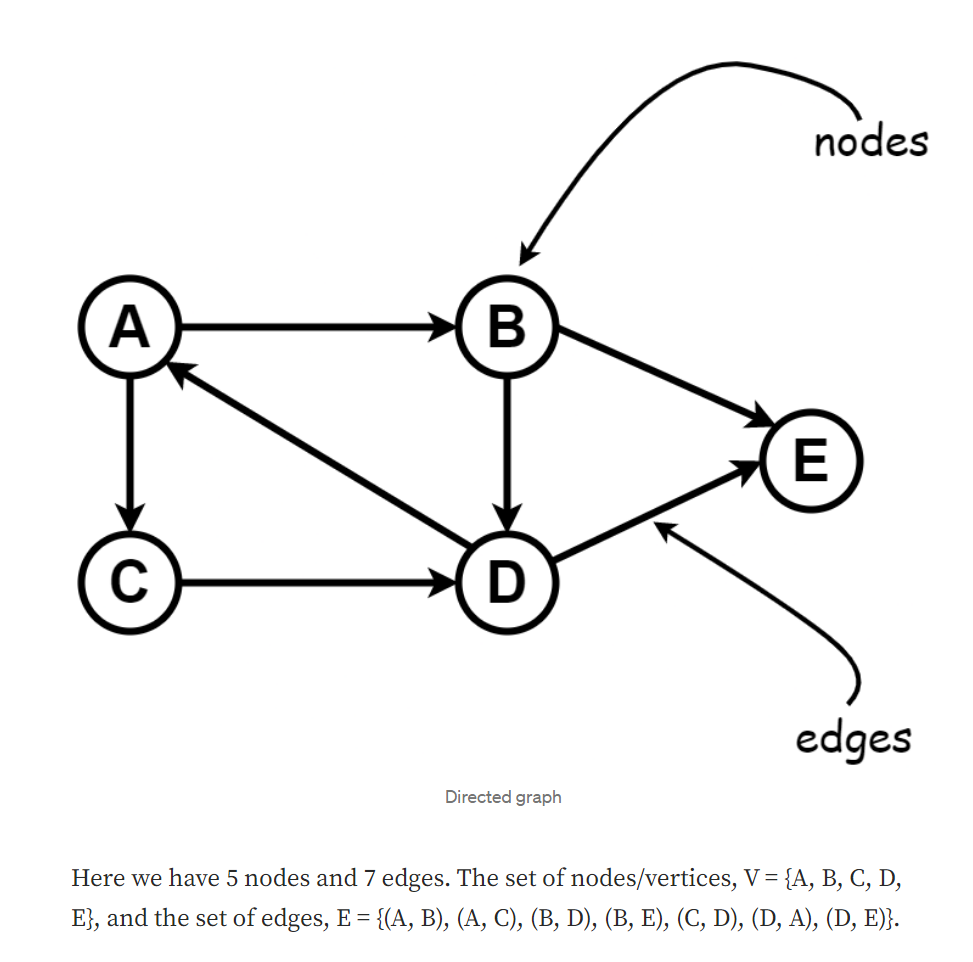
**Graph**

# **The Basics of Graph**

A graph is a non-linear data structure that consists of a set of **nodes** and **edges**. Nodes are also referred to as vertices. An edge is a path that connects two nodes.

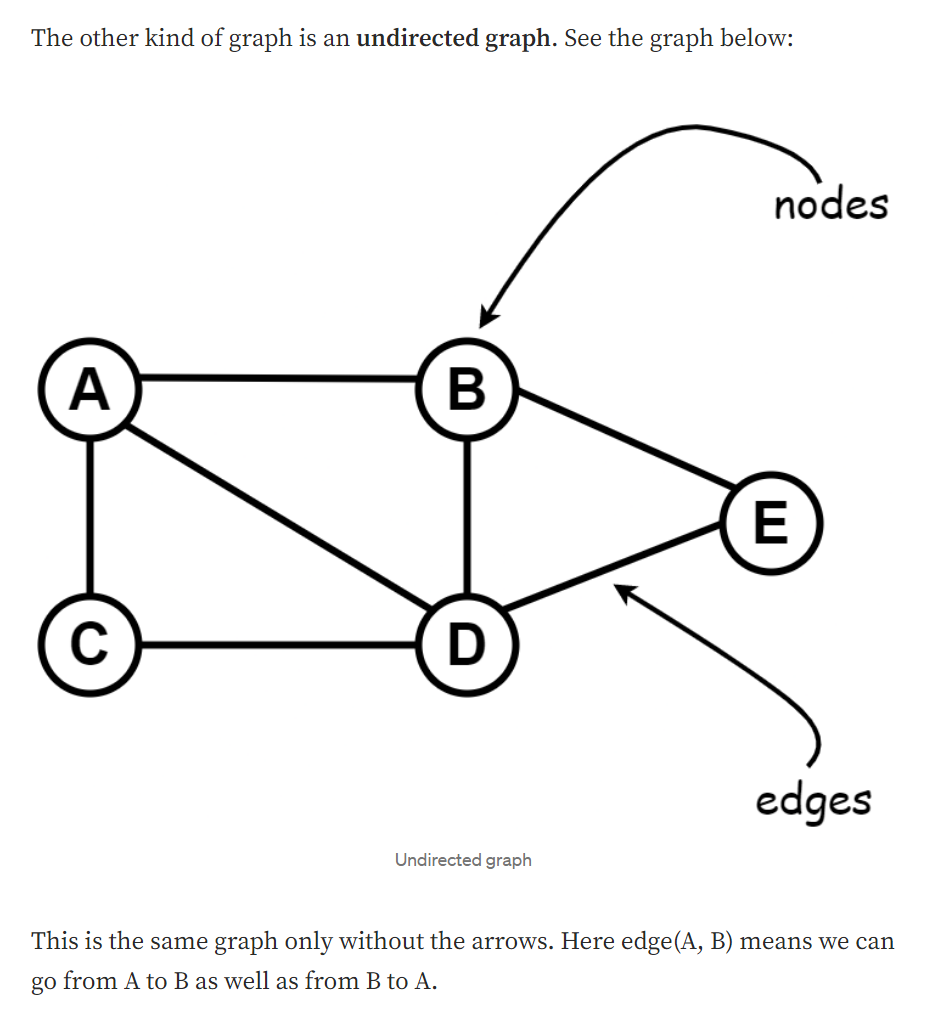
**Types Of Graph :**

1. **Directed Graph :** 
   1. The graph in wich direction is mentioned .
   2. Edges are unidirectional .

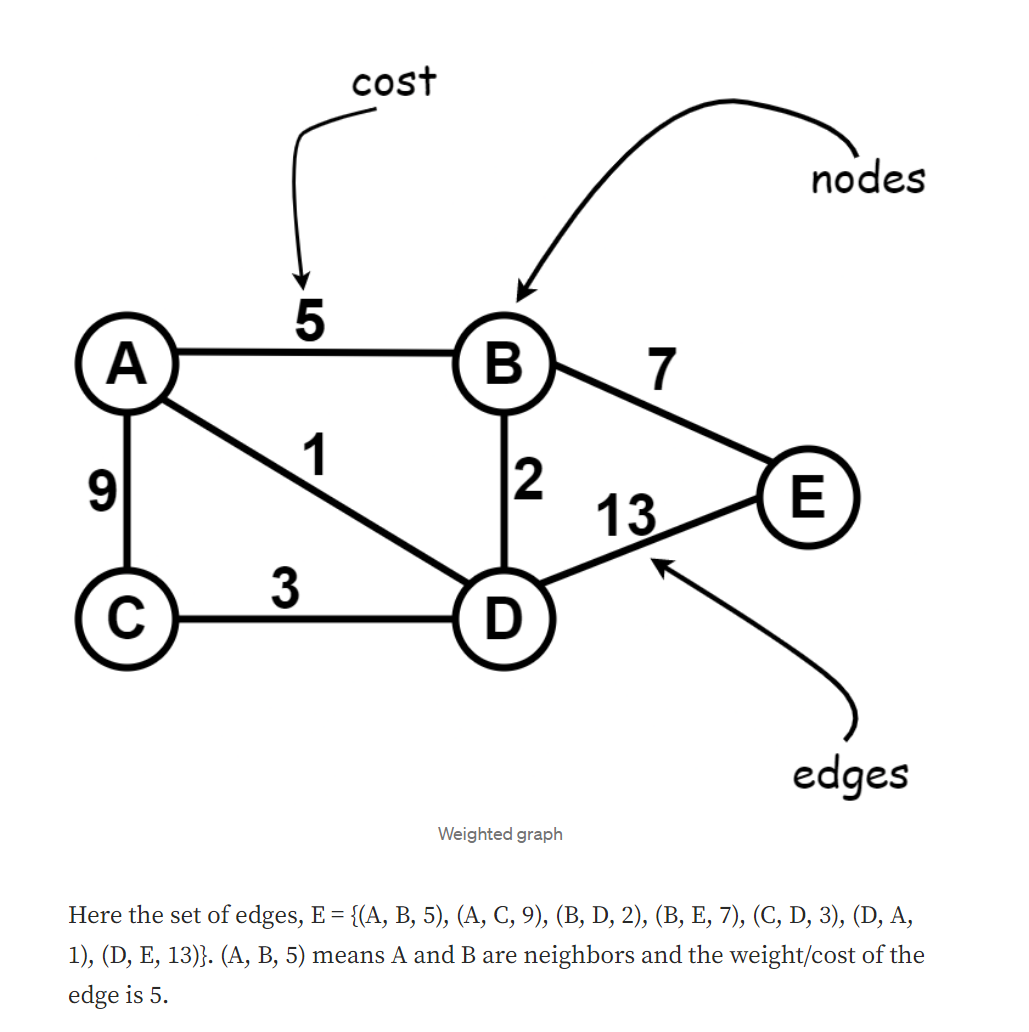


* 1. The set of edges indicates two neighbor nodes. (A, B) means there is a path between node A and node B. The arrow means we can go from node A to node B and not the other way. This kind of graph is called a [**directed graph**](https://en.wikipedia.org/wiki/Directed_graph).

**2. Un-Directed Graph :**

****

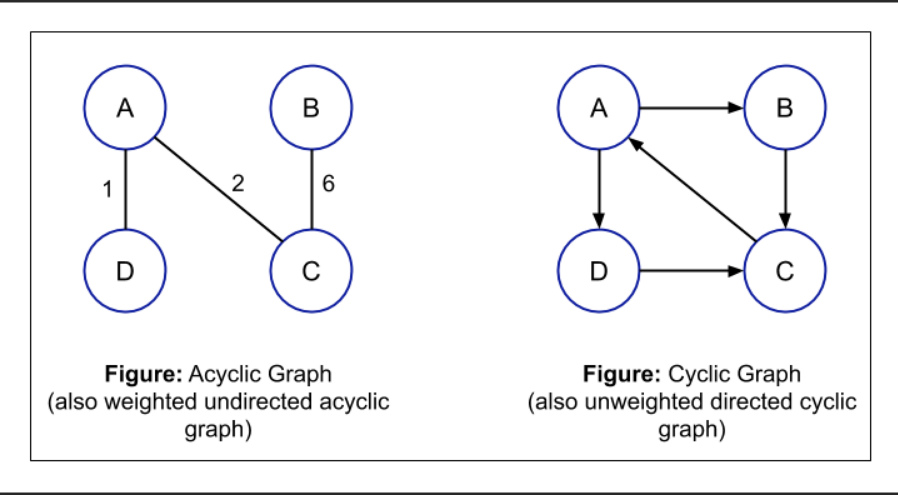
**3. Weighted Graph :**

1. The cost of the paths can also be given in a graph .
2. The cost of an edge is also called weight.
3. 

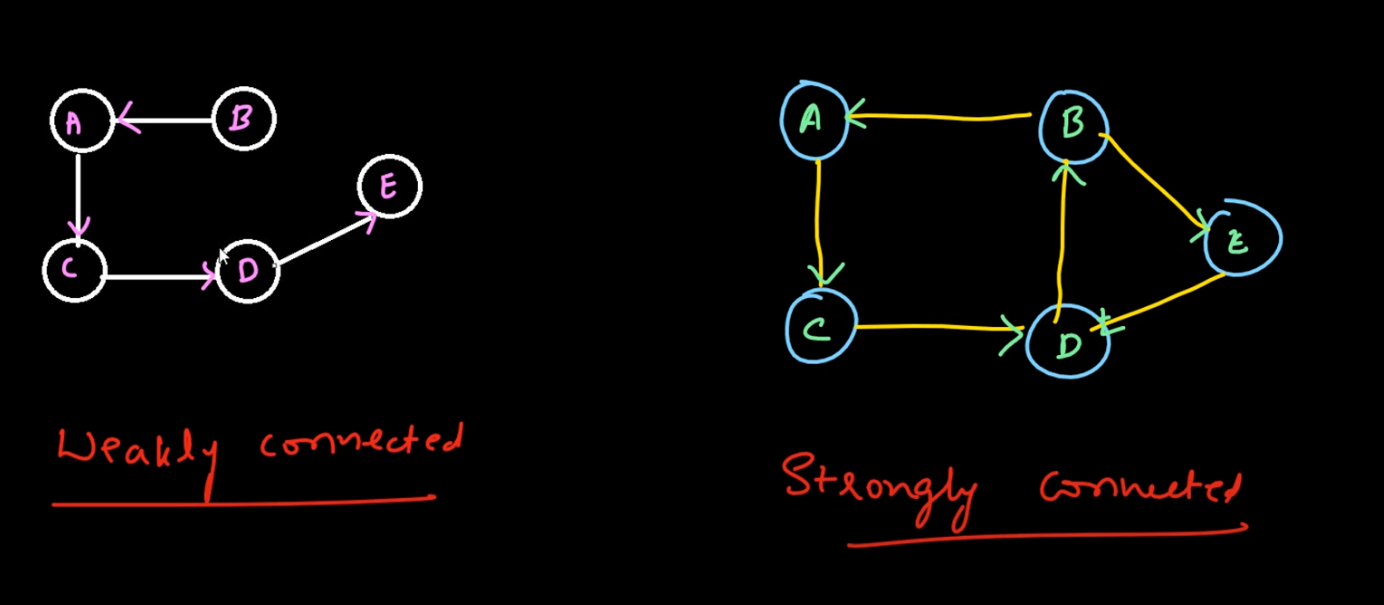
**4. Unweighted Graph :**

1. A graph without the costs actually represents a constant cost .
2. That means the cost of all edges in the graph is the same. It is called an **unweighted graph**.

**6. Cyclic Graph and Acyclic Graph :**

1. A graph is cyclic if a graph contains a path which sart from a vertex and end at the same vertex .
2. 

**Terminology :**

1. **Adjacent Node / Negabour Node :** 
   1. Node x is adjacent to node y if there is an edge between node x to node y
2. **Simple Path :** 
   1. The path is simple if all its vertices are distinct .
3. **Close path :** 
   1. The path is close if the first and last vertices of that are same .
4. **Cycle :** 
   1. In the cycle except first node and last node of path all the other nodes need to be distinct
5. **Strongly and Weekly connected graph** 
   1. 
6. **Degree of Un-directed graph node :**
7. Degree of node = no. of edges connected to it .

**7. Degree of Directed graph node .**

1. Indegree :
   1. Indgree of node = number of edges coming to that node .
2. Outdegree :
   1. Outdegree of node = number of edges coming out from that node .

**8. Complete Graph :**

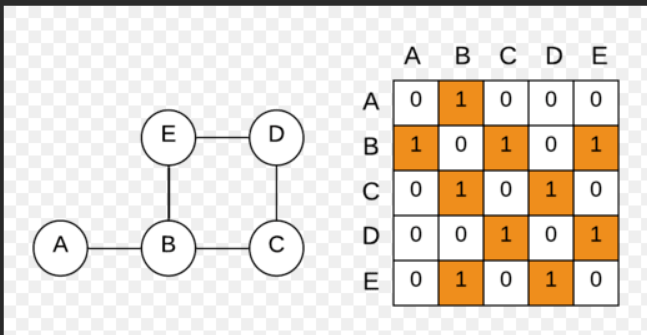
1. If there is edge between every pair of graph is called complete graph .

**Graph Representation**

1. **Adjacency Matrix :**

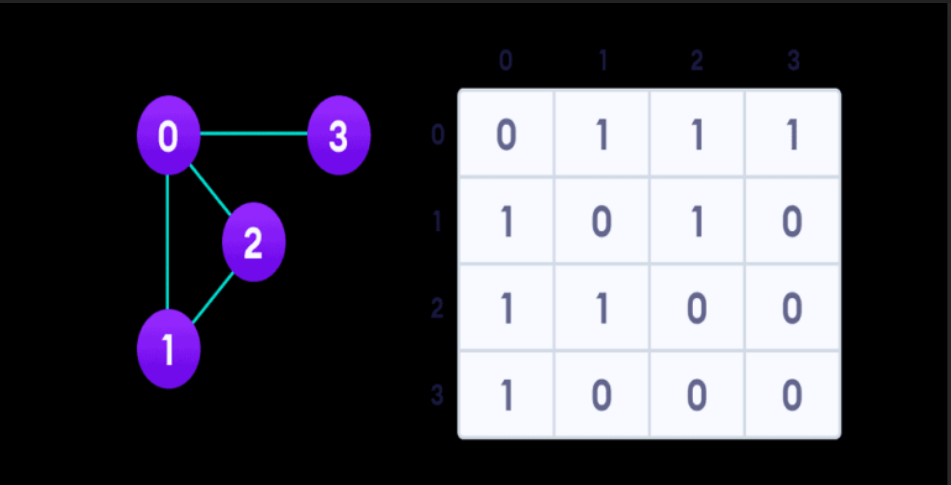
**The adjacency matrix can also be known as the connection matrix. It is a matrix that contains rows and columns which are used to represent a simple labeled graph, with the two numbers 0 or 1 in the position of (Vi, Vj) according to the condition whether the two Vi and Vj are adjacent or not.**

1. Representation of Unweight graph matrix :

****

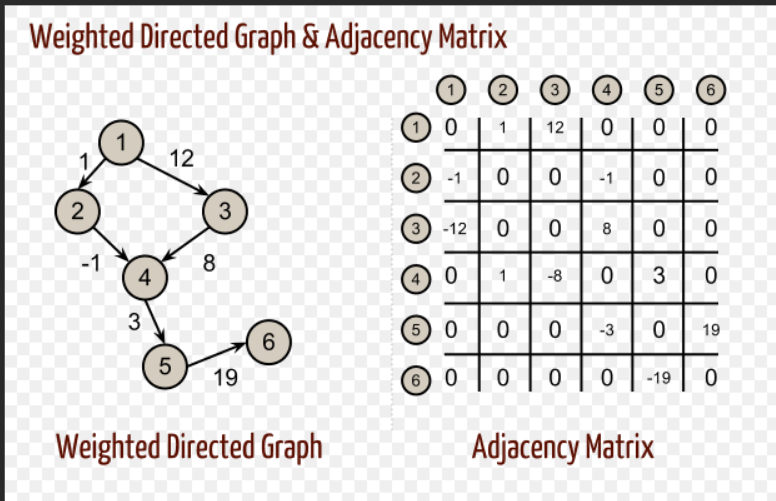
**If the value on the both the side of main diagonals of an Adjacecy matrix are same**

**then the matrix is symmentrix matrix .**

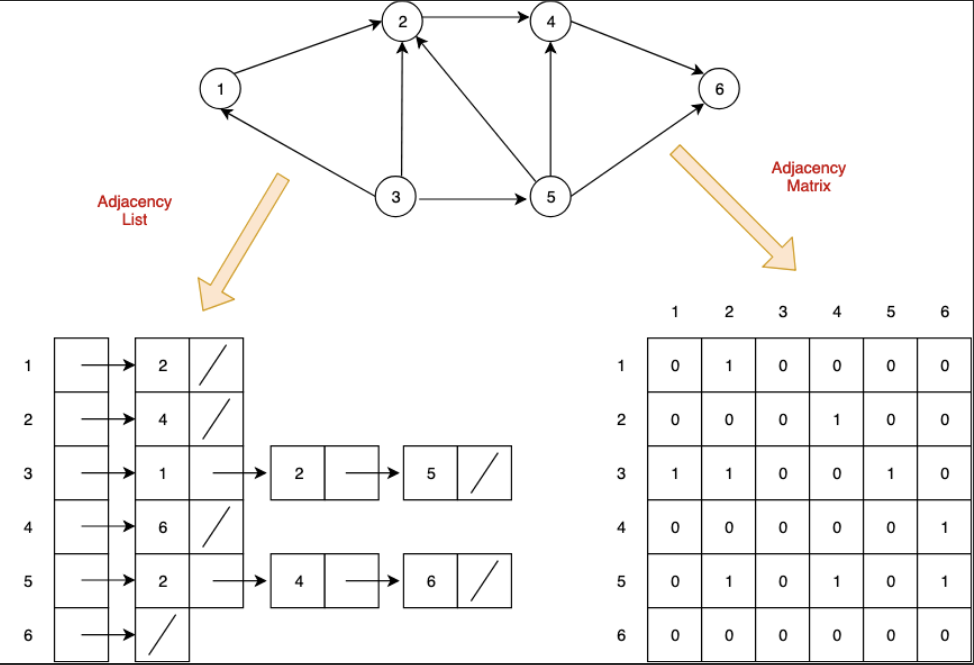
****

**Note :** Symmetric Matrix matrix represent the undirected graph .

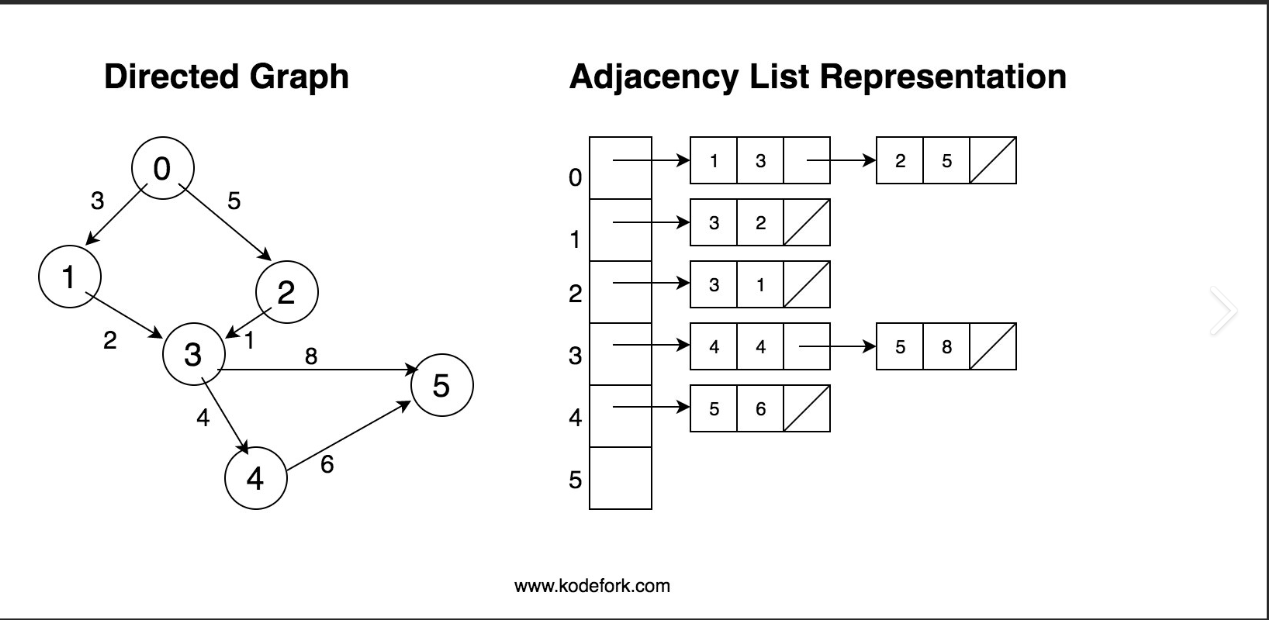
B. Representation of weight graph matrix :



**C.Graph representation : Adjacency list | Directed Graph | Un-weighted**



**D.Graph representation : Adjacency list | Directed Graph | weighted**



# Graph : Adjacency List Representaion

# un-directed graph | un-weighted graph

graph={"a":["b","c"],"b":["a","c"],"c":["a","b"]}

print(graph)

# directed | weighted graph

# using list of List

graph = {"A":[["B",3],["C",4]],"B":[["C",2]],"C":0}

# using dictionary

graph = {"A":{"B":3,"C":4},"B":{"C":2},"C":0}

**Operations On Graph :**

1. **Insertion** 
   1. **directed | Weighted graph**
   2. **Undirected | Un- weighted graph**
2. **Deletion**
3. **Traversal**