[Scan or click here for more resources](#)

What is a Graph? GT 1st

A graph is a pictorial representation of a set of objects where some pairs of objects are connected by links. The interconnected objects are represented by points termed as **vertices**, and the links that connect the vertices are called **edges**.

Formally, 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.

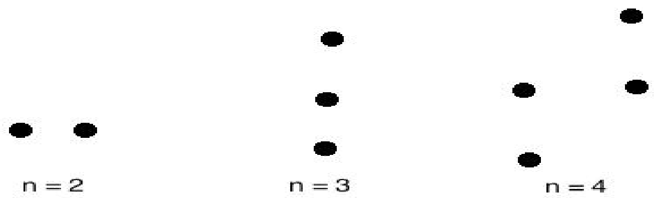
Types of Graphs

Though, there are a lot of different types of graphs depending upon the number of vertices, number of edges, interconnectivity, and their overall structure, some of such common types of graphs are as follows:

1. Null Graph

A **null graph** is a graph in which there are no edges between its vertices. A null graph is also called empty graph.

Example



A null graph with n vertices is denoted by Nn.

2. Trivial Graph

A **trivial graph** is the graph which has only one vertex.

Example

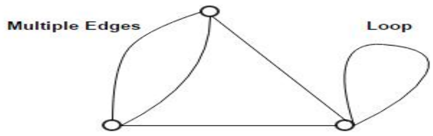
In the above graph, there is only one vertex 'v' without any edge. Therefore, it is a trivial graph.

3. Simple Graph

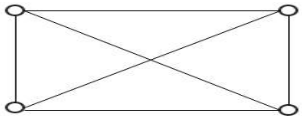
A **simple graph** is the undirected graph with **no parallel edges** and **no loops**.

A simple graph which has n vertices, the degree of every vertex is at most n -1.

Example



Not a Simple Graph

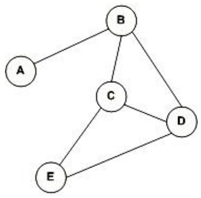

Simple Graph

In the above example, First graph is not a simple graph because it has two edges between the vertices A and B and it also has a loop.

Second graph is a simple graph because it does not contain any loop and parallel edges.

4. Undirected Graph

An **undirected graph** is a graph whose edges are **not directed**.



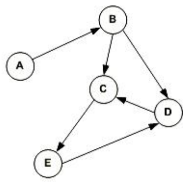
Example

In the above graph since there is no directed edges, therefore it is an undirected graph.

5. Directed Graph

A **directed graph** is a graph in which the **edges are directed** by arrows.

Directed graph is also known as **digraphs**.



Example

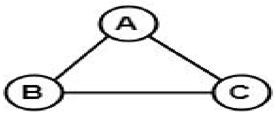
In the above graph, each edge is directed by the arrow. A directed edge has an arrow from A to B, means A is related to B, but B is not related to A.

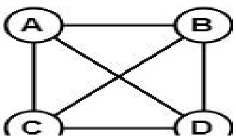
6. Complete Graph

A graph in which every pair of vertices is joined by exactly one edge is called **complete graph**. It contains all possible edges.

A complete graph with n vertices contains exactly nC2 edges and is represented by Kn.

Example



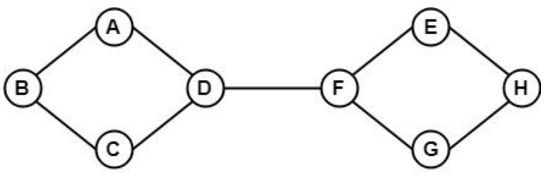


In the above example, since each vertex in the graph is connected with all the remaining vertices through exactly one edge therefore, both graphs are complete graph.

7. Connected Graph

A **connected graph** is a graph in which we can visit from any one vertex to any other vertex. In a connected graph, at least one edge or path exists between every pair of vertices.

Example

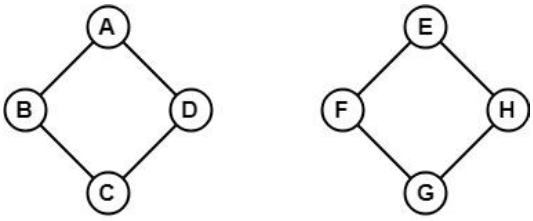


In the above example, we can traverse from any one vertex to any other vertex. It means there exists at least one path between every pair of vertices therefore, it a connected graph.

8. Disconnected Graph

A **disconnected graph** is a graph in which any path does not exist between every pair of vertices.

Example



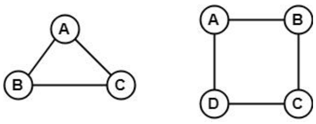
The above graph consists of two independent components which are disconnected. Since it is not possible to visit from the vertices of one component to the vertices of other components therefore, it is a disconnected graph.

9. Regular Graph

A **Regular graph** is a graph in which degree of all the vertices is same.

If the degree of all the vertices is k, then it is called k-regular graph.   Size=q=p*r/2

Example



In the above example, all the vertices have degree 2. Therefore they are called 2- **Regular graph**.

10. Cyclic Graph

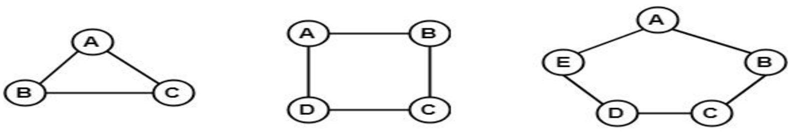
A graph with 'n' vertices (where, n>=3) and 'n' edges forming a cycle of 'n' with all its edges is known as **cycle graph**.

A graph containing at least one cycle in it is known as a **cyclic graph**.

In the cycle graph, degree of each vertex is 2.

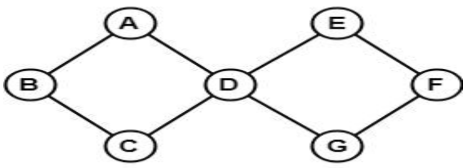
The cycle graph which has n vertices is denoted by Cn.

Example 1



In the above example, all the vertices have degree 2. Therefore they all are cyclic graphs.

Example 2

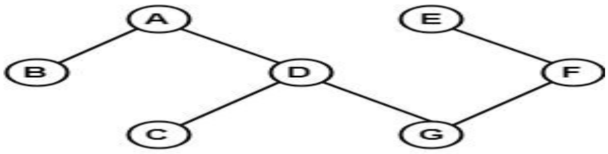


Since, the above graph contains two cycles in it therefore, it is a cyclic graph.

11. Acyclic Graph

A graph which does not contain any cycle in it is called as an **acyclic graph**.

Example



Since, the above graph does not contain any cycle in it therefore, it is an acyclic graph.

12. Bipartite Graph

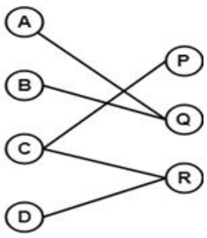
A **bipartite graph** is a graph in which the vertex set can be partitioned into two sets such that edges only go between sets, not within them.

A graph G (V, E) is called bipartite graph if its vertex-set V(G) can be decomposed into two non-empty disjoint subsets V1(G) and V2(G) in such a way that each edge e ∈ E(G) has its one last joint in V1(G) and other last point in V2(G).

The partition V = V1 ∪ V2 is known as bipartition of G.

Example 1

Example 2



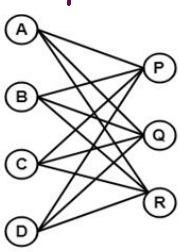
13. Complete Bipartite Graph

A **complete bipartite graph** is a bipartite graph in which each vertex in the first set is joined to each vertex in the second set by exactly one edge.

A complete bipartite graph is a bipartite graph which is complete.

1. Complete Bipartite graph = Bipartite graph + Complete graph

Example



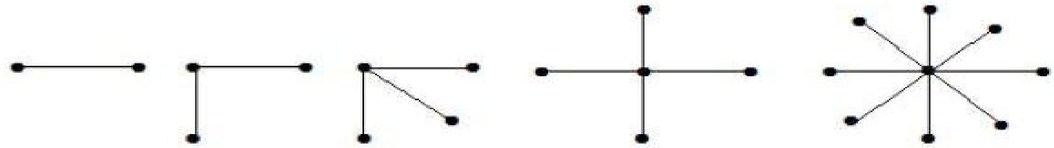
The above graph is known as $K_{4,3}$.

14. Star Graph

A star graph is a complete bipartite graph in which n-1 vertices have degree 1 and a single vertex have degree (n -1). This exactly looks like a star where (n - 1) vertices are connected to a single central vertex.

A star graph with n vertices is denoted by S_n.

Example



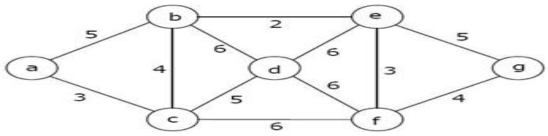
In the above example, out of n vertices, all the (n-1) vertices are connected to a single vertex. Hence, it is a star graph.

15 Weighted Graph

A weighted graph is a graph whose edges have been labeled with some weights or numbers.

The length of a path in a weighted graph is the sum of the weights of all the edges in the path.

Example

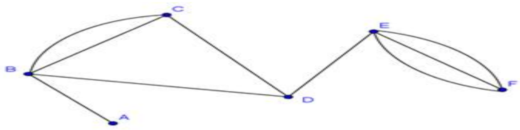


In the above graph, if path is a -> b -> c -> d -> e -> g then the length of the path is 5 + 4 + 5 + 6 + 5 = 25.

16. Multi-graph

A graph in which there are multiple edges between any pair of vertices or there are edges from a vertex to itself (loop) is called a **multi - graph**.

Example

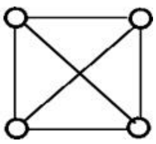


In the above graph, vertex-set B and C are connected with two edges. Similarly, vertex sets E and F are connected with 3 edges. Therefore, it is a multi graph.

17. Planar Graph

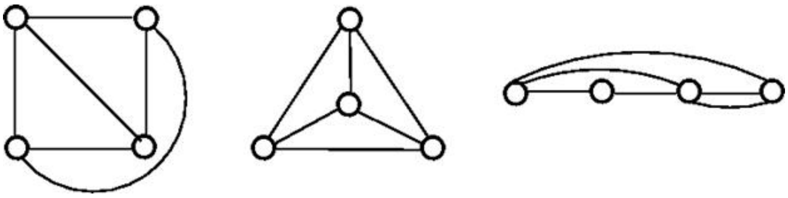
A **planar graph** is a graph that we can draw in a plane in such a way that no two edges of it cross each other except at a vertex to which they are incident.

Example



The above graph may not seem to be planar because it has edges crossing each other. But we can redraw the above graph.

The three plane drawings of the above graph are:

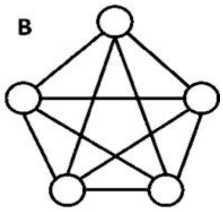


The above three graphs do not consist of two edges crossing each other and therefore, all the above graphs are planar.

18. Non - Planar Graph

A graph that is not a planar graph is called a non-planar graph. In other words, a graph that cannot be drawn without at least on pair of its crossing edges is known as non-planar graph.

Example



The above graph is a non - planar graph.

* A **pseudograph** G = (V, E) consists of a set V of vertices, a set E of edges, and a function g from E to {{u, v} : u, v ∈ V }. An edge if a loop if f(e) = {u, u} = {u} for some u ∈ V .

Handshaking Theory

We can also call the handshaking theory as the Sum of degree theorem or Handshaking Lemma. The handshaking theory states that the sum of degree of all the vertices for a graph will be double the number of edges contained by that graph. The symbolic representation of handshaking theory is described as follows:
Here,

$$\sum_{i=1}^{n} d(v_i) = 2 \times |E|$$

'd' is used to indicate the degree of the vertex.
'v' is used to indicate the vertex.
'e' is used to indicate the edges.

Handshaking Theorem:

There are some conclusions in the handshaking theorem, which must be drawn, which are described as follows:
In any graph:

- There must be even numbers for the sum of degree of all the vertices.
- If there are odd degrees for all the vertices, then the sum of degree of these vertices must always remain even.
- If there are some vertices that have an odd degree, then the number of these vertices will be even.

Graph Representations

In graph theory, a graph representation is a technique to store graph into the memory of computer.
To represent a graph, we just need the set of vertices, and for each vertex the neighbors of the vertex (vertices which is directly connected to it by an edge). If it is a weighted graph, then the weight will be associated with each edge.
There are different ways to optimally represent a graph, depending on the density of its edges, type of operations to be performed and ease of use.

1. Adjacency Matrix

- Adjacency matrix is a sequential representation.

- It is used to represent which nodes are adjacent to each other. i.e. is there any edge connecting nodes to a graph.
- In this representation, we have to construct a nXn matrix A. If there is any edge from a vertex i to vertex j, then the corresponding element of A, $a^{i,j}$ = 1, otherwise $a^{i,j}$= 0.

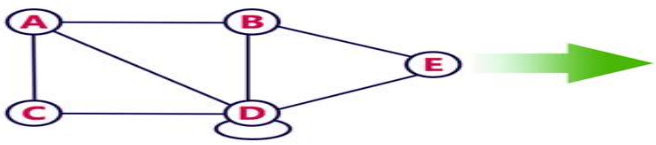
Note, even if the graph on 100 vertices contains only 1 edge, we still have to have a 100x100 matrix with lots of zeroes.

- If there is any weighted graph then instead of 1s and 0s, we can store the weight of the edge.

Example

Consider the following **undirected graph representation**:

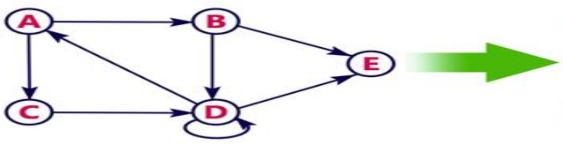
Undirected graph representation



$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Directed graph represenation

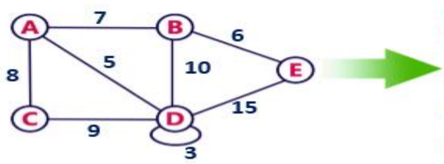
See the directed graph representation:



$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

In the above examples, 1 represents an edge from row vertex to column vertex, and 0 represents no edge from row vertex to column vertex.

Undirected weighted graph represenation



$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{pmatrix} 0 & 7 & 8 & 5 & 0 \\ 7 & 0 & 0 & 10 & 6 \\ 8 & 0 & 0 & 9 & 0 \\ 5 & 10 & 9 & 3 & 15 \\ 0 & 6 & 0 & 15 & 0 \end{pmatrix} \end{array}$$

Pros: Representation is easier to implement and follow.

Cons: It takes a lot of space and time to visit all the neighbors of a vertex, we have to traverse all the vertices in the graph, which takes quite some time.

2. Incidence Matrix

In **Incidence matrix representation**, graph can be represented using a matrix of size:

Total number of vertices by total number of edges.

It means if a graph has 4 vertices and 6 edges, then it can be represented using a matrix of 4X6 class. In this matrix, columns represent edges and rows represent vertices.

This matrix is filled with either **0 or 1** or -1. Where,

- 0 is used to represent row edge which is not connected to column vertex.
- 1 is used to represent row edge which is connected as outgoing edge to column vertex.

- -1 is used to represent row edge which is connected as incoming edge to column vertex.

Example

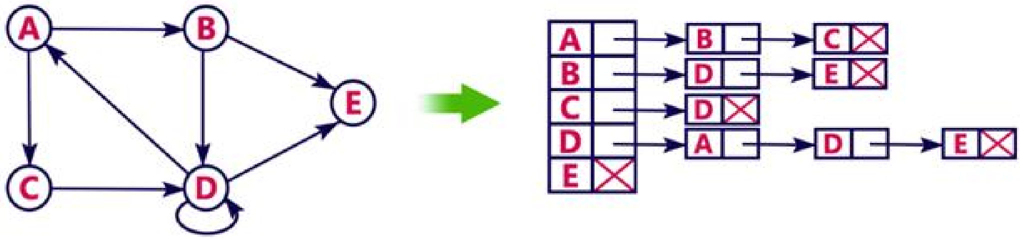
Consider the following directed graph representation.

3. Adjacency List

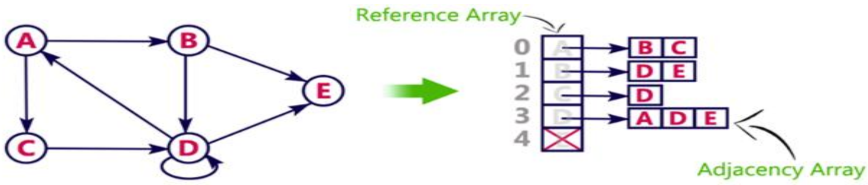
- Adjacency list is a linked representation.
- In this representation, for each vertex in the graph, we maintain the list of its neighbors. It means, every vertex of the graph contains list of its adjacent vertices.
- We have an array of vertices which is indexed by the vertex number and for each vertex v, the corresponding array element points to a **singly linked list** of neighbors of v.

Example

Let's see the following directed graph representation implemented using linked list:



We can also implement this representation using array as follows:



Pros:

- Adjacency list saves lot of space.
- We can easily insert or delete as we use linked list.
- Such kind of representation is easy to follow and clearly shows the adjacent nodes of node.

Cons:

- The adjacency list allows testing whether two vertices are adjacent to each other but it is slower to support this operation.

Isomorphisms of graphs :

The simple graphs G1 = (V1, E1) and G2 = (V2, E2) are isomorphic if there is a one-to-one and onto function f from V1 to V2 with the property that a and b are adjacent in G1 if and only if f(a) and f(b) are adjacent in G2, for all a and b in V1. Such a function is called an isomorphism both have — such a property is called an invariant with respect to isomorphism of simple graphs:

1. isomorphic simple graphs must have the same number of vertices
2. isomorphic simple graphs must have the same number of edges
3. the degrees of the vertices in isomorphic simple graphs must be the same.

Hovel hakimi :

The arrangement of degree o all vertices of a graph in non increasing .

7 6 5 4 4 3 2 1

6 6 6 6 3 3 2 2

7 6 6 4 4 3 2 2

8 7 7 6 4 2 1 1

Connectivity :

In graph theory, connectivity refers to the degree to which a graph is connected, or the ease with which one can traverse the graph from one vertex to another.

Walk: A walk is a sequence of vertices and edges that starts at one vertex and ends at another vertex. The vertices in the sequence are connected by the edges, but the edges may be repeated or visited multiple times. A walk may also have loops, which are edges that connect a vertex to itself.

Open walk :- when a walk begins and end vertex are not same then called a open walk.

Closed walk:-when a walk begins and end at the same vertex it is called closed walk.

Path: A path is a walk that does not have any repeated vertices or edges. In other words, a path is a sequence of vertices and edges that starts at one vertex and ends at another vertex, but does not revisit any vertex or edge along the way. A path is sometimes called a simple path to distinguish it from a path that has repeated vertices or edges.

Circuit: A circuit is a walk that starts and ends at the same vertex. A circuit may also be called a closed walk or a cycle. A circuit may or may not have repeated vertices or edges.

Rank & Nullity of a Graph :

Let G(V,E) be a graph with n vertices & m edges and K Components.

i.e.; $|G(V)| = n$ & $|G(E)| = m$ we define the rank P(G) & nullity $\mu(G)$ as follows :

If G Is not connected :

P(G) = Rank of G = n - k

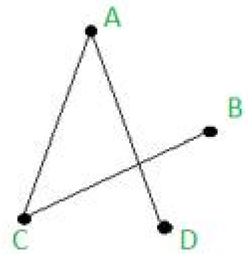
μ(G) = Nullity of G = m - n + k

If G Is connected :

P(G) = Rank of G = n - 1

μ(G) = Nullity of G = m - n + 1

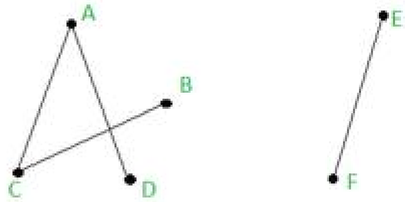
Example 1 :The Graph shown below is connected :



Graph G

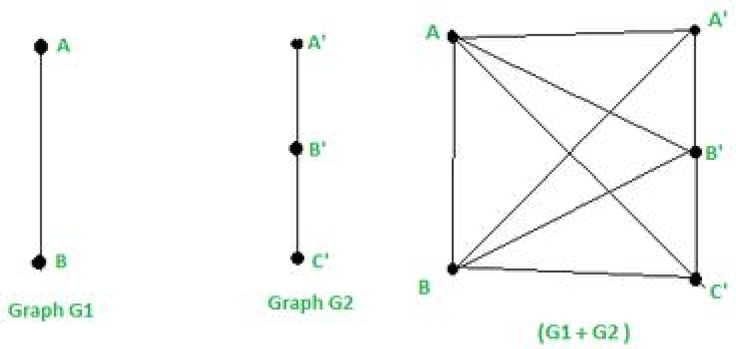
$|G(V)| = n = 4$ &
$|G(E)| = m = 3$
$P(G)$ = Rank of $G = n - 1 = 4 - 1 = 3$
$\mu(G)$ = Nullity of $G = m - n + 1 = 3 - 4 + 1 = 0$

Example 2 : The Graph shown below is not connected :



$|G(V)| = n = 6$ &
$|G(E)| = m = 4$ &
No of components = $k = 2$
$P(G)$ = Rank of $G = n - k = 6 - 2 = 4$
$\mu(G)$ = Nullity of $G = m - n + k = 4 - 6 + 2 = 0$

Addition Of 2 Graphs :
If we have 2 graphs, G_1 & G_2 such that their vertices intersection is null ($V(G_1) \cap V(G_2) = \emptyset$) , then the sum :
$G_1 + G_2$ is defined as the graph whose vertex set $V(G_1+G_2)$ is $V(G_1) + V(G_2)$ and the edge set consists of these edges, which are inG_1 & in G_2 & the edges contained by joining each vertex of G_1 to each vertex of G_2.
Example : The addition of 2 graphs shown G_1 & G_2 are :

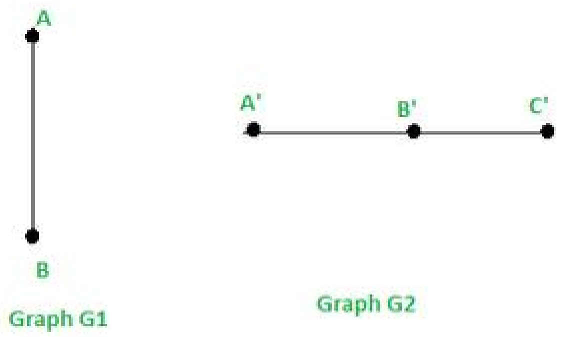


Here : $V(G_1) \cap V(G_2) = \emptyset$

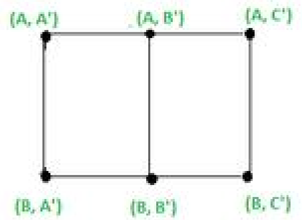
The already contained edges in G1 are, E(G1) : {{A, B}} and the vertices are : V(G1) = {A, B}
The already contained edges in G2 are, E(G2) : {{A', B'} ,{B',C'} }and the vertices are : V(G2) = {A', B', C'}
So the graph , G1 + G2 will have
(i) vertices as : V(G1+G2) = V(G1) + V(G2) = { A, B. A', B', C'}
(ii) and E(G1 + G2) = E(G1) + E(G2) + edges contained by joining each vertex of G1 to each vertex of G2 =
{ {A, B}, {A', B'} ,{B',C'} , {A,A'} , {A, B'}, {A, C'}, {B,A'} , {B, B'}, {B, C'}}

Product of 2 Graphs :

Example : Consider 2 Graphs, G1 & G2 such that :
V(G1) = {A, B}
E(G1) = { {A, B} }
V(G2) = {A', B', C'}
E(G2) ={ {A', B'} ,{B',C'} }



Then Graph G1*G2 will have :



*Graph Product G1 * G2*

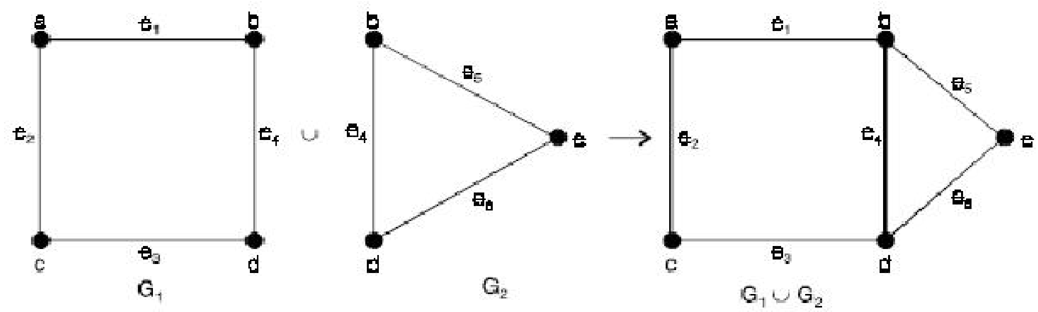
Union :

Given two graphs G_1 and G_2, their union will be a graph such that

$$V(G_1 \cup G_2) = V(G_1) \cup V(G_2)$$

and

$$E(G_1 \cup G_2) = E(G_1) \cup E(G_2)$$



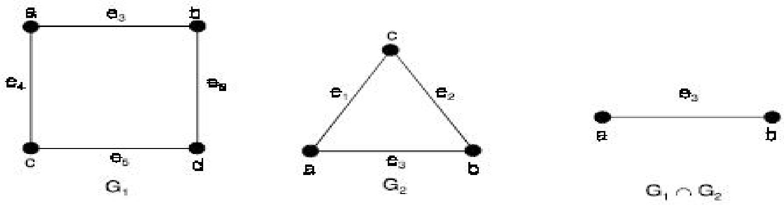
Intersection:

Given two graphs G_1 and G_2 with at least one vertex in common then their intersection will be a graph such that

$$V(G_1 \cap G_2) = V(G_1) \cap V(G_2)$$

and

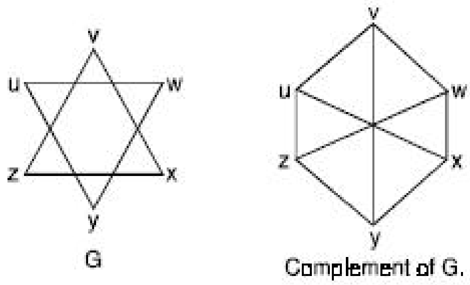
$$E(G_1 \cap G_2) = E(G_1) \cap E(G_2)$$



Complement:

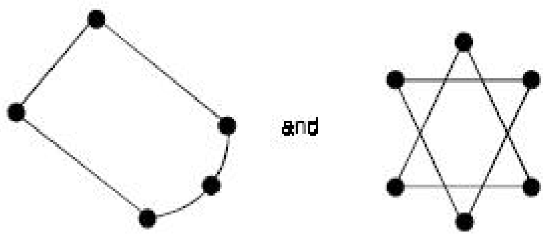
The complement G′ of G is defined as a simple graph with the same vertex set as G and where two vertices *u* and *v* adjacent only when they are not adjacent in G.

For example,

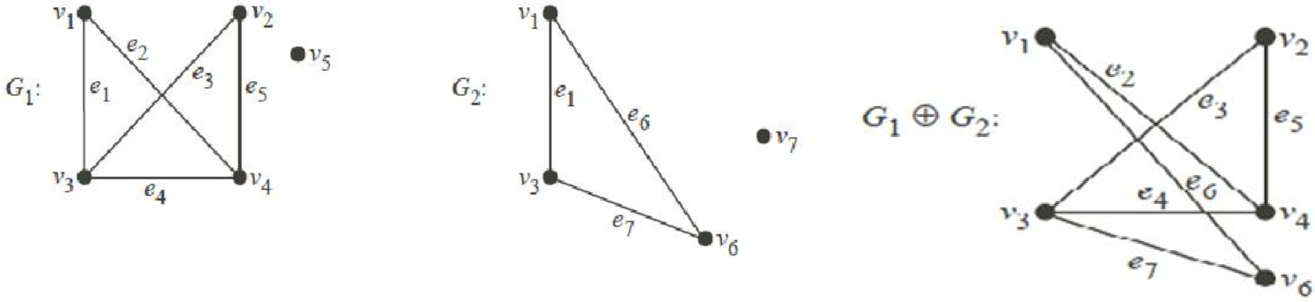


A graph G is self-complementary if it is isomorphic to its complement.

For example, the graphs



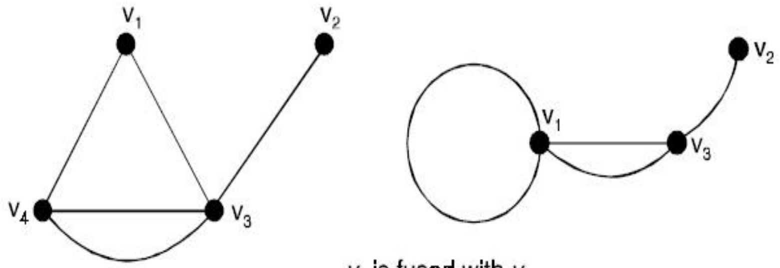
Ring Sum:



Fusion:

A pair of vertices v_1 and v_2 in graph G is said to be 'fused' if these two vertices are replaced by a single new vertex v such that every edge that was adjacent to either v_1 or v_2 or both is adjacent v.

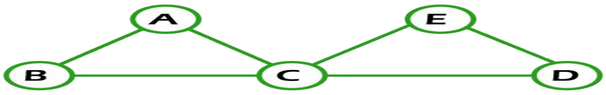
Thus we observe that the fusion of two vertices does not alter the number of edges of graph but reduced the vertices by one.



v_1 is fused with v_4

Euler Graph

If all the vertices of any connected graph have an even degree, then this type of graph will be known as the Euler graph. In other words, we can say that an Euler graph is a type of connected graph which have the Euler circuit. The simple example of Euler graph is described as follows:



Example of Euler Graph

The above graph is a connected graph, and the vertices of this graph contain the even degree. Hence we can say that this graph is an Euler graph.

Euler Path

We can also call the Euler path as Euler walk or Euler Trail. The definition of Euler trail and Euler walk is described as follows:

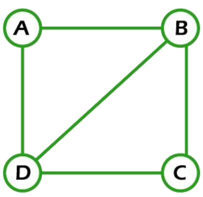
- *If there is a connected graph with a trail that has all the edges of the graph, then that type of trail will be known as the Euler trail.*
- *If there is a connected graph, which has a walk that passes through each and every edge of the graph only once, then that type of walk will be known as the Euler walk.*

Note: If more than two vertices of the graph contain the odd degree, then that type of graph will be known as the Euler Path.

Examples of Euler path:

There are a lot of examples of the Euler path, and some of them are described as follows:

Example 1: *In the following image, we have a graph with 4 nodes. Now we have to determine whether this graph contains an Euler path.*



Solution:

The above graph will contain the Euler path if each edge of this graph must be visited exactly once, and the vertex of this can be repeated. So if we begin our path from vertex B and then go to vertices C, D, B, A, and D, then in this process, each and every edge is visited exactly once, and it also contains repeated vertex. So the above graph contains an Euler path, which is described as follows:

Euler path : BCDBAD

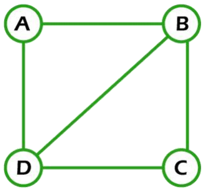
Semi Euler Graph

If there is a connected graph that does not have an Euler circuit, but it has an Euler trail, then that type of graph will be known as the semi-Euler graph. Any graph will be known as semi Euler graph if it satisfies two conditions, which are described as follows:

- For this, the graph must be connected
- This graph must contain an Euler trail

Example of Semi-Euler graph

In this example, we have a graph with 4 nodes. Now we have to determine whether this graph is a semi-Euler graph.



Semi - Euler Graph

Solution:

Here,

- There is an Euler trail in this graph, i.e., BCDBAD.
- But there is no Euler circuit.
- Hence, this graph is a semi-Euler graph.

Hamiltonian Path

In a connected graph, if there is a walk that passes each and every vertex of a graph only once, this walk will be known as the Hamiltonian path. In this walk, the edges should not be repeated. There is one more definition to describe the Hamiltonian path: if a connected graph contains a Path with all the vertices of the graph, this type of path will be known as the Hamiltonian path.

Hamilton path/cycle: A Hamilton path in a graph is a path that visits every vertex exactly once. A Hamilton cycle is a cycle that visits every vertex exactly once, and starts and ends at the same vertex. A graph is said to be Hamiltonian if it contains a Hamilton cycle.

Hamiltonian Circuit

In a connected graph, if there is a walk that passes each and every vertex of the graph only once and after completing the walk, return to the starting vertex, then this type of walk will be known as a Hamiltonian circuit. For the Hamiltonian circuit, there must be no repeated edges. We can also be called Hamiltonian circuit as the Hamiltonian cycle.

There are some more definitions of the Hamiltonian circuit, which are described as follows:

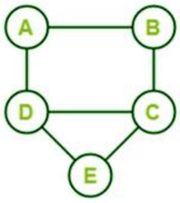
- If there is a Hamiltonian path that begins and ends at the same vertex, then this type of cycle will be known as a Hamiltonian circuit.

- In the connected graph, if there is a cycle with all the vertices of the graph, this type of cycle will be known as a Hamiltonian circuit.
- A closed Hamiltonian path will also be known as a Hamiltonian circuit.

Examples of Hamiltonian Circuit

There are a lot of examples of the Hamiltonian circuit, which are described as follows:

Example 1: In the following graph, we have 5 nodes. Now we have to determine whether this graph contains a Hamiltonian circuit.

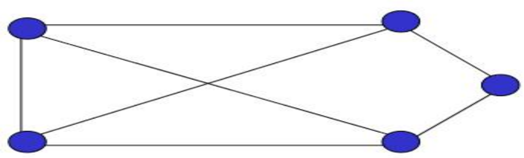


Solution: =

The above graph contains the Hamiltonian circuit if there is a path that starts and ends at the same vertex. So when we start from the A, then we can go to B, C, E, D, and then A. So this is the path that contains all the vertices (A, B, C, D, and E) only once, except the starting vertex, and there is no repeating edge. That's why we can say that this graph has a Hamiltonian circuit, which is described as follows:

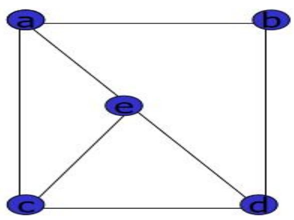
Hamiltonian circuit = ABCDEA

DIRAC'S Theorem: if G is a simple graph with n vertices with $n \geq 3$ such that the degree of every vertex in G is at least n/2 then G has a Hamilton circuit.



n=5
D(v)=5/2

- **ORE'S Theorem:** if G is a simple graph with n vertices with $n \geq 3$ such that deg (u) + deg (v) $\geq$ n for every pair of nonadjacent vertices u and v in G, then G has a Hamilton circuit.



ad=6
be=5
bc=5
da=6

Problem 1.37. *How many vertices and how many edges do the following graphs have ?*

(*i*) K_n (*ii*) C_n (*iii*) W_n (*iv*) $K_{m,\,n}$ (v) Q_n.

Solution. (*i*) n vertices and $\dfrac{n(n-1)}{2}$ edges.

(*ii*) n vertices and n edges

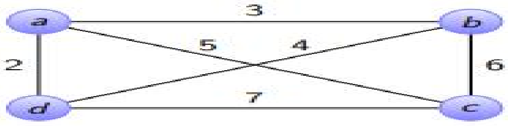
(*iii*) $n + 1$ vertices and $2n$ edges

(*iv*) $m + n$ vertices and mn edges

(v) 2^n vertices and $n \,.\, 2^{n-1}$ edges.

The Traveling Salesman Problem

For example, consider all possible cycles (a circuit that only visits each vertex once, except the start/end vertex) in our weighted K_4.



cycle	weight
$a\ b\ c\ d\ a$	$3+6+7+2=18$
$a\ b\ d\ c\ a$	$3+4+7+5=19$
$a\ c\ b\ d\ a$	$5+6+4+2=17$
$a\ c\ d\ b\ a$	$5+7+4+3=19$
$a\ d\ b\ c\ a$	$2+4+6+5=17$
$a\ d\ c\ b\ a$	$2+7+6+3=18$

- There are six cycles – but half of these are reversals of the other half.
- The starting point is arbitrary – a given cycle will have the same cost regardless of starting point.