Graphs-

- A graph is a collection of vertices connected to each other through a set of edges.
- The study of graphs is known as Graph Theory.

Formal Definition

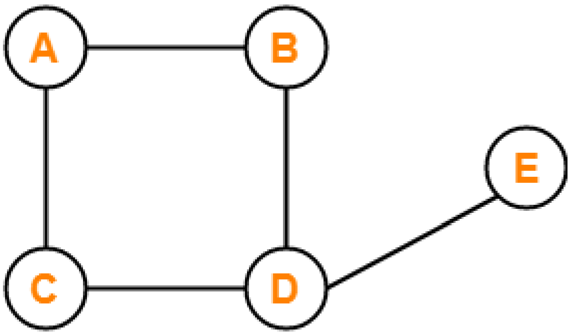
Formally,

A graph is defined as an ordered pair of a set of vertices and a set of edges.

G = (V, E)

Here, V is the set of vertices and E is the set of edges connecting the vertices.

Example-



Example of Graph

In this graph,

V = { A , B , C , D , E }

E = { AB , AC , BD , CD , DE }

Types of Graphs-

Various important types of graphs in graph theory are

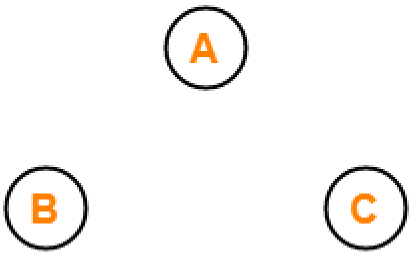
1. Null Graph
2. Trivial Graph

3. Non-directed Graph
4. Directed Graph
5. Connected Graph
6. Disconnected Graph
7. Regular Graph
8. Complete Graph
9. Cycle Graph
10. Cyclic Graph
11. Acyclic Graph
12. Finite Graph
13. Infinite Graph
14. Bipartite Graph
15. Planar Graph
16. Simple Graph
17. Multi Graph
18. Pseudo Graph
19. Euler Graph
20. Hamiltonian Graph

1. Null Graph-

- A graph whose edge set is empty is called as a null graph.
- In other words, a null graph does not contain any edges in it.

Example-



Example of Null Graph

Here,

- This graph consists only of the vertices and there are no edges in it.
- Since the edge set is empty, therefore it is a null graph.

2. Trivial Graph-

- A graph having only one vertex in it is called as a trivial graph.
- It is the smallest possible graph.

Example-

Example of Trivial Graph

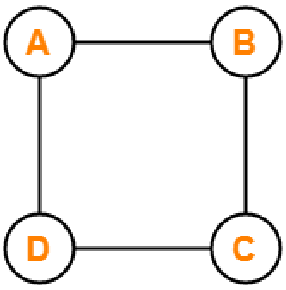
Here,

- This graph consists of only one vertex and there are no edges in it.
- Since only one vertex is present, therefore it is a trivial graph.

3. Non-Directed Graph-

- A graph in which all the edges are undirected is called as a non-directed graph.
- In other words, edges of an undirected graph do not contain any direction.

Example-



Example of Non-Directed Graph

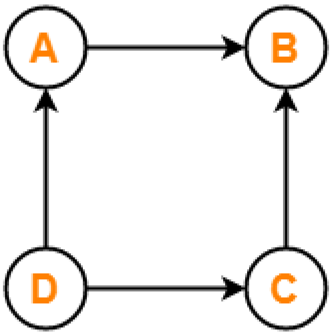
Here,

- This graph consists of four vertices and four undirected edges.
- Since all the edges are undirected, therefore it is a non-directed graph.

4. Directed Graph-

- A graph in which all the edges are directed is called as a directed graph.
- In other words, all the edges of a directed graph contain some direction.
- Directed graphs are also called as digraphs.

Example-



Example of Directed Graph

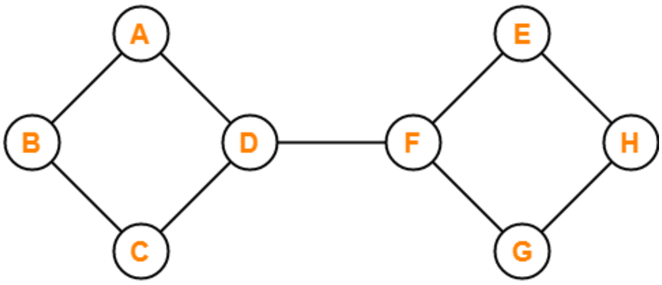
Here,

- This graph consists of four vertices and four directed edges.
- Since all the edges are directed, therefore it is a directed graph.

5. Connected Graph-

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

Example-



Example of Connected Graph

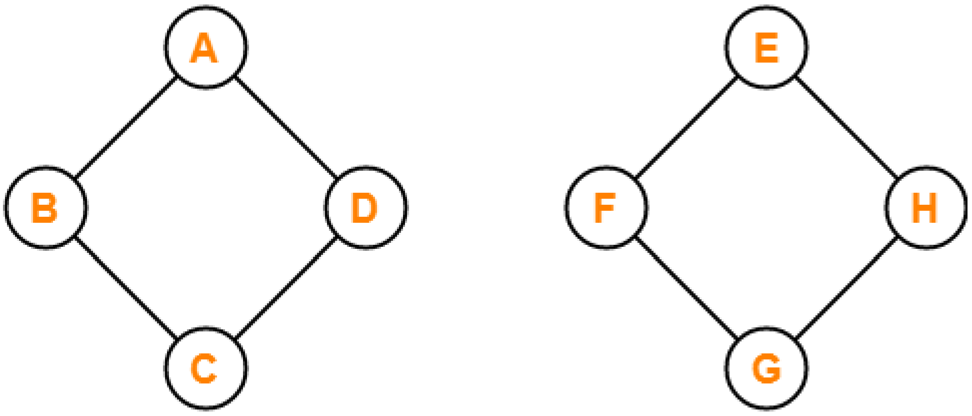
Here,

- In this graph, we can visit from any one vertex to any other vertex.
- There exists at least one path between every pair of vertices.
- Therefore, it is a connected graph

6. Disconnected Graph-

- A graph in which there does not exist any path between at least one pair of vertices is called as a disconnected graph.

Example-



Example of Disconnected Graph

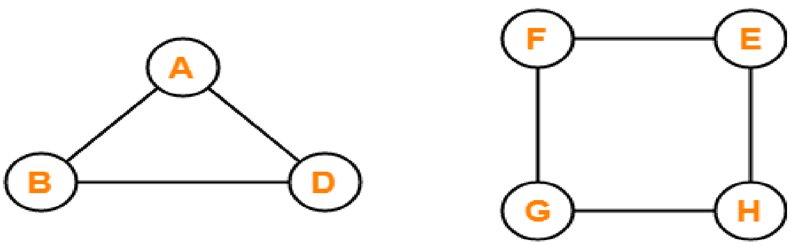
Here,

- This graph consists of two independent components which are disconnected.
- It is not possible to visit from the vertices of one component to the vertices of other component.
- Therefore, it is a disconnected graph.

7. Regular Graph-

- A graph in which degree of all the vertices is same is called as a regular graph.
- If all the vertices in a graph are of degree 'k', then it is called as a "k-regular graph".

Examples-



Examples of Regular Graph

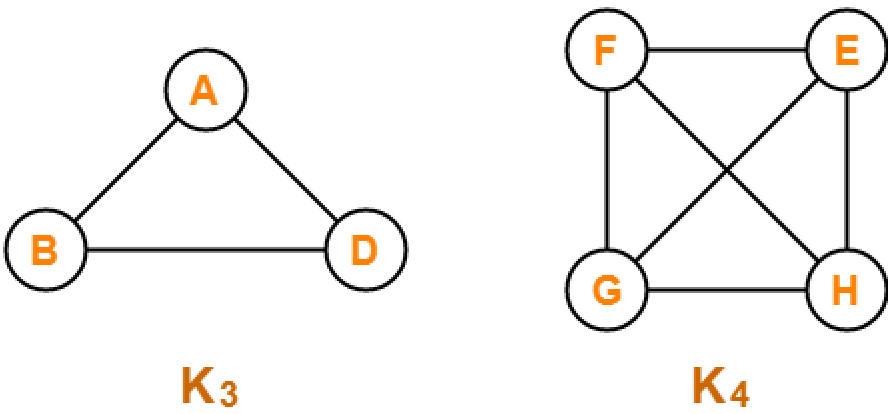
In these graphs,

- All the vertices have degree-2.
- Therefore, they are 2-Regular graphs.

8. Complete Graph-

- A graph in which exactly one edge is present between every pair of vertices is called as a complete graph.
- A complete graph of 'n' vertices contains exactly ${}^nC_2$ edges.
- A complete graph of 'n' vertices is represented as K_n.

Examples-



Examples of Complete Graph

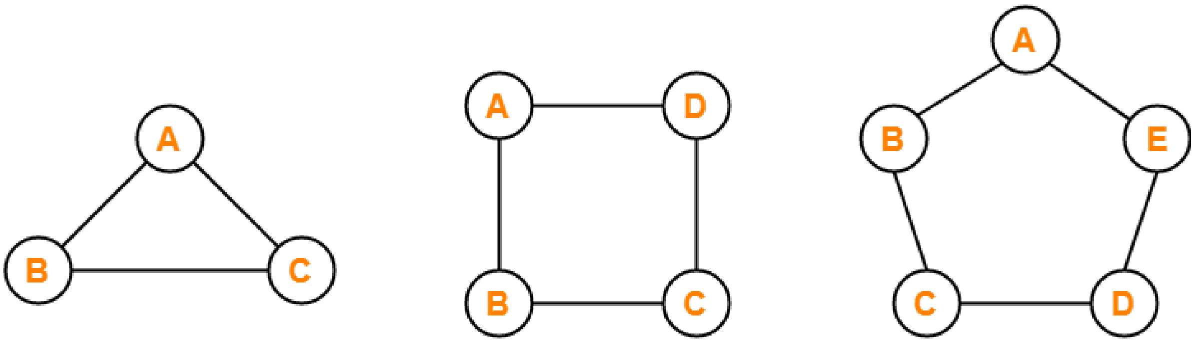
In these graphs,

- Each vertex is connected with all the remaining vertices through exactly one edge.
- Therefore, they are complete graphs.

9. Cycle Graph-

- A simple graph of 'n' vertices (n>=3) and n edges forming a cycle of length 'n' is called as a cycle graph.
- In a cycle graph, all the vertices are of degree 2.

Examples-



Examples of Cycle Graph

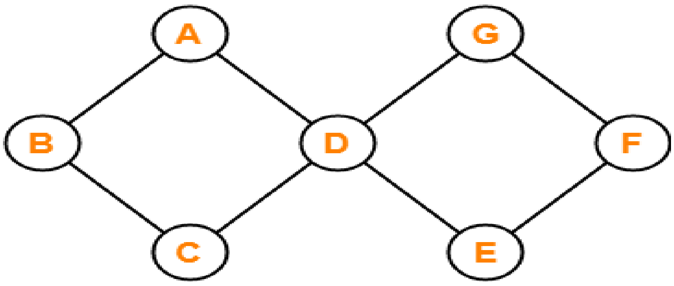
In these graphs,

- Each vertex is having degree 2.
- Therefore, they are cycle graphs.

10. Cyclic Graph-

- A graph containing at least one cycle in it is called as a cyclic graph.

Example-



Example of Cyclic Graph

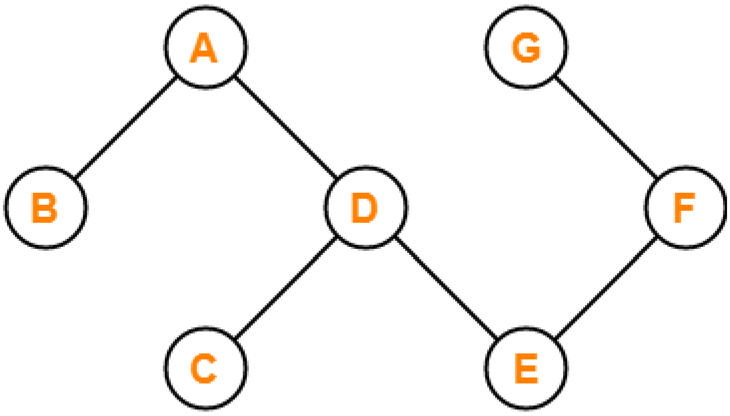
Here,

- This graph contains two cycles in it.
- Therefore, it is a cyclic graph.

11. Acyclic Graph-

- A graph not containing any cycle in it is called as an acyclic graph.

Example-



Example of Acyclic Graph

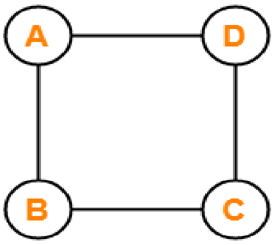
Here,

- This graph do not contain any cycle in it.
- Therefore, it is an acyclic graph.

12. Finite Graph-

- A graph consisting of finite number of vertices and edges is called as a finite graph.

Example-



Example of Finite Graph

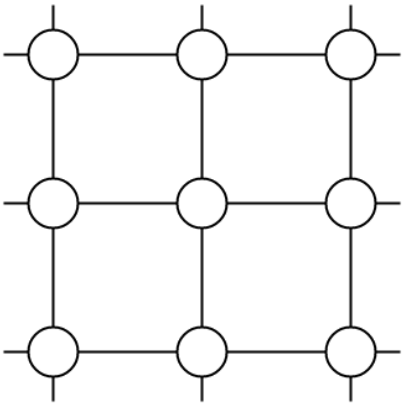
Here,

- This graph consists of finite number of vertices and edges.
- Therefore, it is a finite graph.

13. Infinite Graph-

- A graph consisting of infinite number of vertices and edges is called as an infinite graph.

Example-



Example of Infinite Graph

Here,

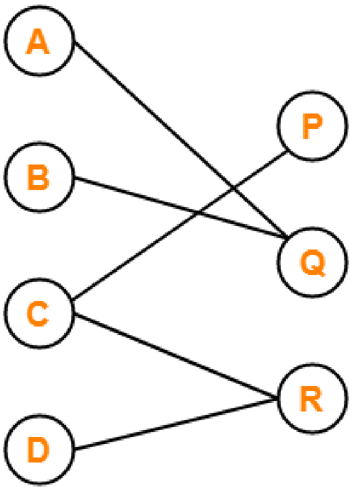
- This graph consists of infinite number of vertices and edges.
- Therefore, it is an infinite graph.

14. Bipartite Graph-

A bipartite graph is a graph where-

- Vertices can be divided into two sets X and Y.
- The vertices of set X only join with the vertices of set Y.
- None of the vertices belonging to the same set join each other.

Example-

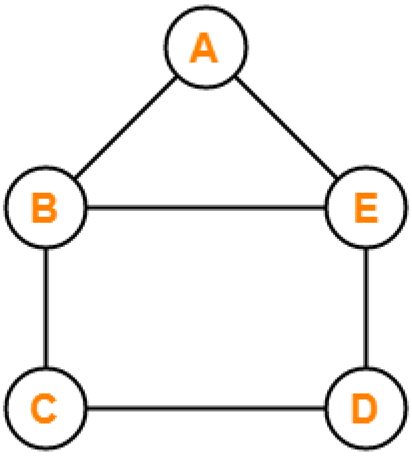


Example of Bipartite Graph

<u>15. Planar Graph-</u>

- A planar graph is a graph that we can draw in a plane such that no two edges of it cross each other.

<u>Example-</u>



Example of Planar Graph

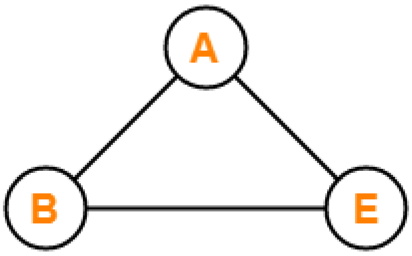
Here,

- This graph can be drawn in a plane without crossing any edges.
- Therefore, it is a planar graph.

<u>16. Simple Graph-</u>

- A graph having no self loops and no parallel edges in it is called as a simple graph.

<u>Example-</u>



Example of Simple Graph

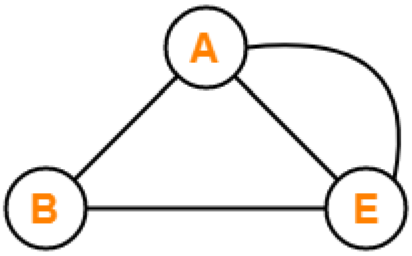
Here,

- This graph consists of three vertices and three edges.
- There are neither self loops nor parallel edges.
- Therefore, it is a simple graph.

17. Multi Graph-

- A graph having no self loops but having parallel edge(s) in it is called as a multi graph.

Example-



Example of Multi Graph

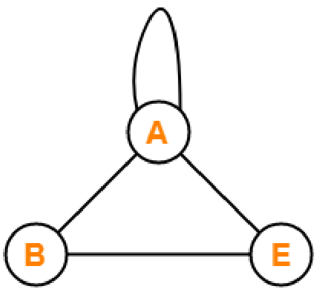
Here,

- This graph consists of three vertices and four edges out of which one edge is a parallel edge.
- There are no self loops but a parallel edge is present.
- Therefore, it is a multi graph.

18. Pseudo Graph-

- A graph having no parallel edges but having self loop(s) in it is called as a pseudo graph.

Example-



Example of Pseudo Graph

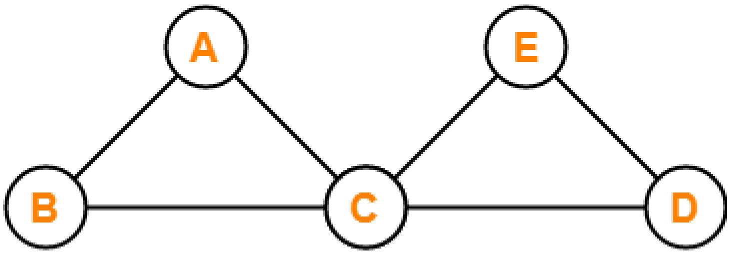
Here,

- This graph consists of three vertices and four edges out of which one edge is a self loop.
- There are no parallel edges but a self loop is present.
- Therefore, it is a pseudo graph.

19. Euler Graph-

- Euler Graph is a connected graph in which all the vertices are even degree.

Example-



Example of Euler Graph

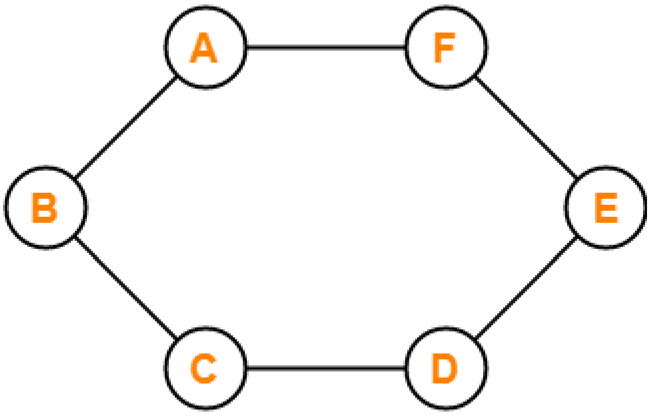
Here,

- This graph is a connected graph.
- The degree of all the vertices is even.
- Therefore, it is an Euler graph.

20. Hamiltonian Graph-

- If there exists a closed walk in the connected graph that visits every vertex of the graph exactly once (except starting vertex) without repeating the edges, then such a graph is called as a Hamiltonian graph.

Example-



Example of Hamiltonian Graph

Here,

- This graph contains a closed walk ABCDEFG that visits all the vertices (except starting vertex) exactly once.
- All the vertices are visited without repeating the edges.
- Therefore, it is a Hamiltonian Graph.

Important Points-

- Edge set of a graph can be empty but vertex set of a graph can not be empty.
- Every polygon is a 2-Regular Graph.
- Every complete graph of 'n' vertices is a (n-1)-regular graph.
- Every regular graph need not be a complete graph.

Remember-

The following table is useful to remember different types of graphs-

	Self-Loop(s)	Parallel Edge(s)
Graph	Yes	Yes
Simple Graph	No	No
Multi Graph	No	Yes
Pseudo Graph	Yes	No

Degree of Vertices

It is the number of vertices adjacent or connected to a vertex V.

Notation − deg(V).

In a simple graph with n number of vertices, the degree of any vertices is −

$\deg(v) = n - 1 \ \forall\, v \in G$

A vertex can form an edge with all other vertices except by itself. So the degree of a vertex will be up to the **number of vertices in the graph minus 1**. This 1 is for the self-vertex as it cannot form a loop by itself. If there is a loop at any of the vertices, then it is not a Simple Graph.

Degree of vertex can be considered under two cases of graphs −

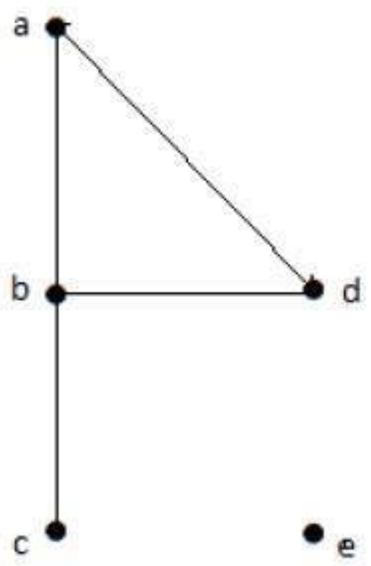
- Undirected Graph
- Directed Graph

Degree of Vertex in an Undirected Graph

An undirected graph has no directed edges. Consider the following examples.

Example 1

Take a look at the following graph −



In the above Undirected Graph,

- deg(a) = 2, as there are 2 edges meeting at vertex 'a'.
- deg(b) = 3, as there are 3 edges meeting at vertex 'b'.
- deg(c) = 1, as there is 1 edge formed at vertex 'c'

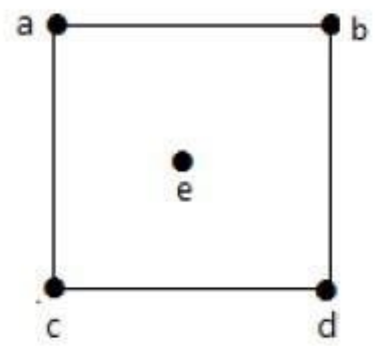
So 'c' is a **pendent vertex**.

- deg(d) = 2, as there are 2 edges meeting at vertex 'd'.
- deg(e) = 0, as there are 0 edges formed at vertex 'e'.

So 'e' is an **isolated vertex**.

Example 2

Take a look at the following graph −



In the above graph,

deg(a) = 2, deg(b) = 2, deg(c) = 2, deg(d) = 2, and deg(e) = 0.

The vertex 'e' is an isolated vertex. The graph does not have any pendent vertex.

Degree of Vertex in a Directed Graph

In a directed graph, each vertex has an **indegree** and an **outdegree**.

Indegree of a Graph

- Indegree of vertex V is the number of edges which are coming into the vertex V.
- **Notation** − $deg^{-}(V)$.

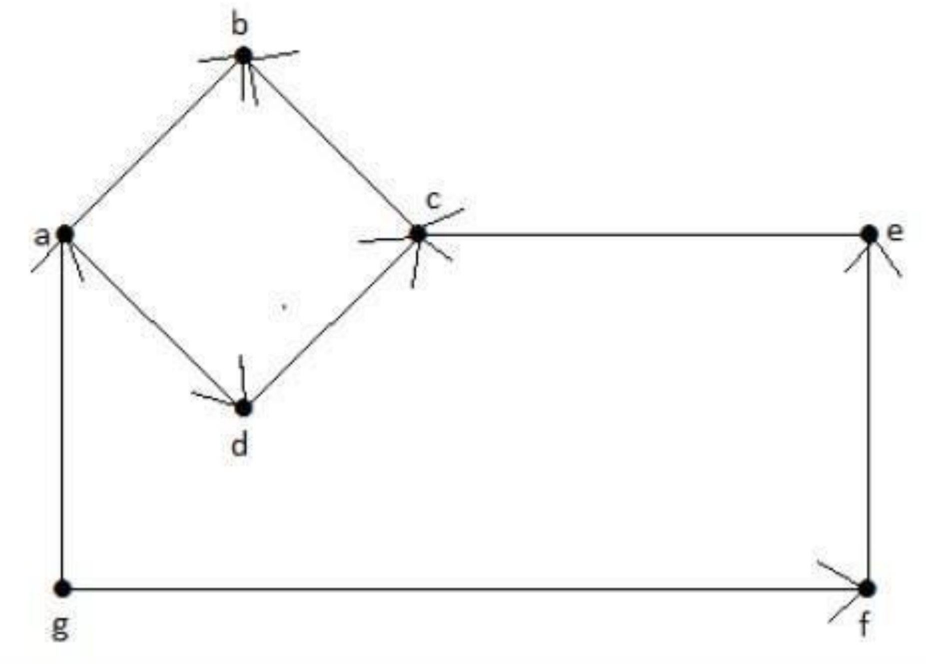
Outdegree of a Graph

- Outdegree of vertex V is the number of edges which are going out from the vertex V.
- **Notation** − $deg^+(V)$.

Consider the following examples.

Example 1

Take a look at the following directed graph. Vertex 'a' has two edges, 'ad' and 'ab', which are going outwards. Hence its outdegree is 2. Similarly, there is an edge 'ga', coming towards vertex 'a'. Hence the indegree of 'a' is 1.



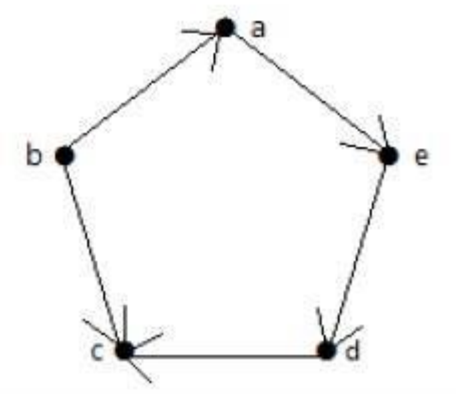
The indegree and outdegree of other vertices are shown in the following table −

Vertex	Indegree	Outdegree
a	1	2
b	2	0
c	2	1

Vertex	Indegree	Outdegree
d	1	1
e	1	1
f	1	1
g	0	2

Example 2

Take a look at the following directed graph. Vertex 'a' has an edge 'ae' going outwards from vertex 'a'. Hence its outdegree is 1. Similarly, the graph has an edge 'ba' coming towards vertex 'a'. Hence the indegree of 'a' is 1.



The indegree and outdegree of other vertices are shown in the following table −

Vertex	Indegree	Outdegree
a	1	1
b	0	2
c	2	0

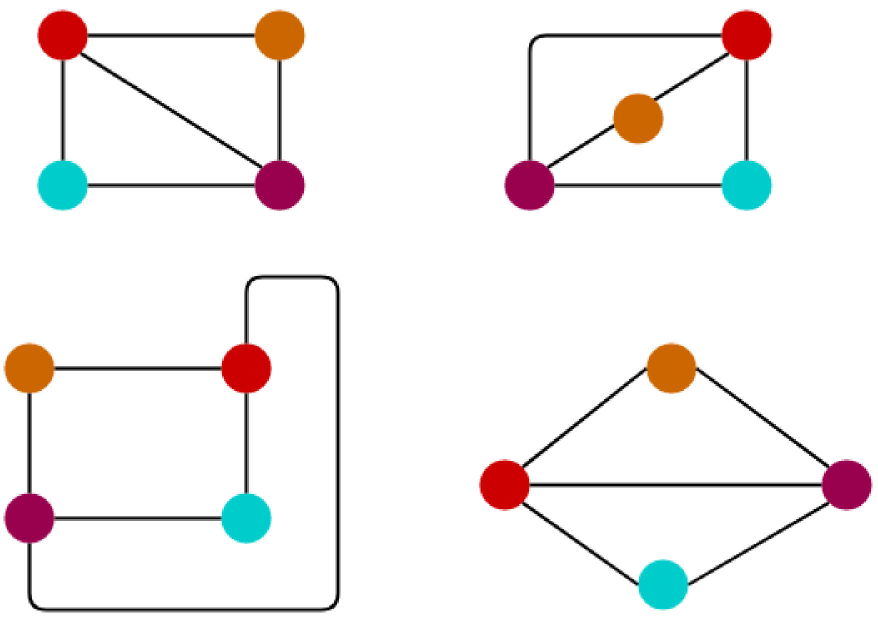
Vertex	Indegree	Outdegree
d	1	1
e	1	1

Graph Isomorphism-

Graph Isomorphism is a phenomenon of existing the same graph in more than one forms.

Such graphs are called as **Isomorphic graphs**.

Graph Isomorphism Example-



Graph Isomorphism Example

Here,

- The same graph exists in multiple forms.
- Therefore, they are **Isomorphic graphs**.

Graph Isomorphism Conditions-

For any two graphs to be isomorphic, following 4 conditions must be satisfied-

1. Number of vertices in both the graphs must be same.
2. Number of edges in both the graphs must be same.
3. Degree sequence of both the graphs must be same.
4. If a cycle of length k is formed by the vertices { v_1 , v_2 , ….. , v_k } in one graph, then a cycle of same length k must be formed by the vertices { $f(v_1)$, $f(v_2)$, ….. , $f(v_k)$ } in the other graph as well.

Degree Sequence

Degree sequence of a graph is defined as a sequence of the degree of all the vertices in ascending order.

Important Points-

- The above 4 conditions are just the necessary conditions for any two graphs to be isomorphic.
- They are not at all sufficient to prove that the two graphs are isomorphic.
- If all the 4 conditions satisfy, even then it can't be said that the graphs are surely isomorphic.
- However, if any condition violates, then it can be said that the graphs are surely not isomorphic.

Sufficient Conditions-

The following conditions are the sufficient conditions to prove any two graphs isomorphic.

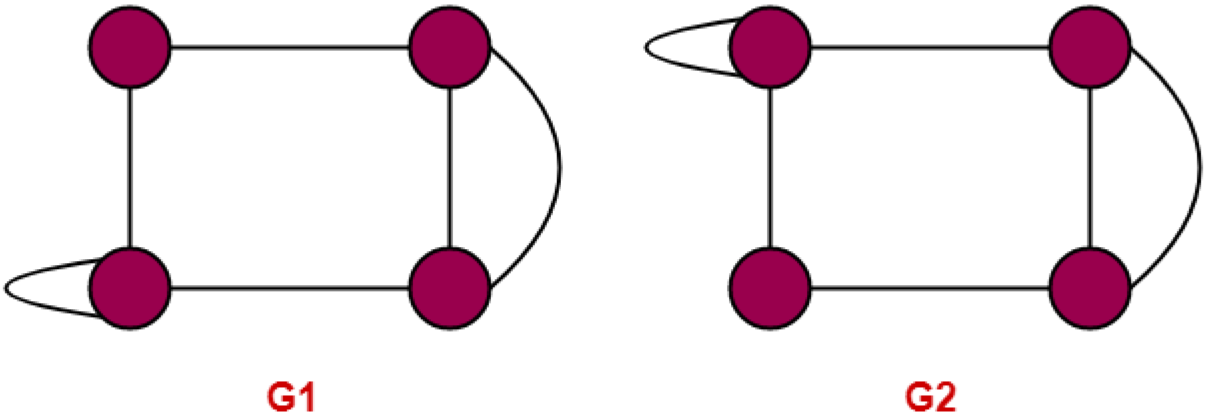
If any one of these conditions satisfy, then it can be said that the graphs are surely isomorphic

- Two graphs are isomorphic if and only if their complement graphs are isomorphic.
- Two graphs are isomorphic if their adjacency matrices are same.
- Two graphs are isomorphic if their corresponding sub-graphs obtained by deleting some vertices of one graph and their corresponding images in the other graph are isomorphic.

PRACTICE PROBLEMS BASED ON GRAPH ISOMORPHISM-

Problem-01:

Are the following two graphs isomorphic?



Solution-

Checking Necessary Conditions-

Condition-01:

- Number of vertices in graph G1 = 4
- Number of vertices in graph G2 = 4

Here,

- Both the graphs G1 and G2 have same number of vertices.
- So, Condition-01 satisfies.

Condition-02:

- Number of edges in graph G1 = 5
- Number of edges in graph G2 = 6

Here,

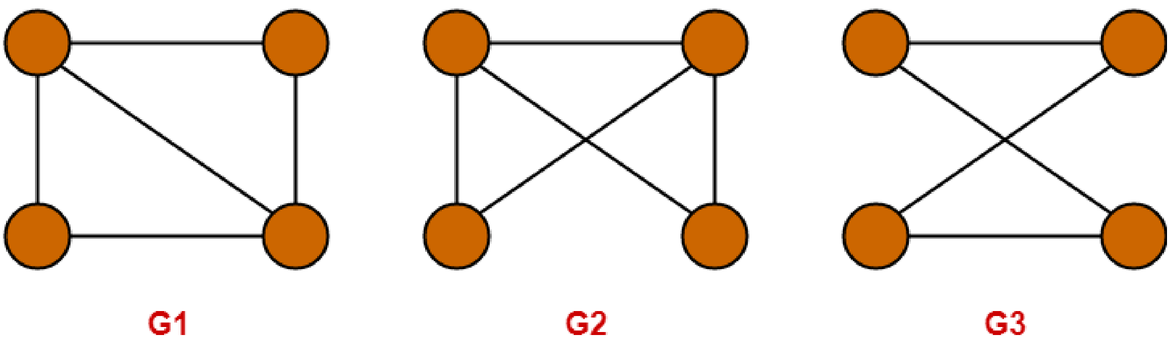
- Both the graphs G1 and G2 have different number of edges.
- So, Condition-02 violates.

Since Condition-02 violates, so given graphs can not be isomorphic.

∴ G1 and G2 are not isomorphic graphs

Problem-02:

Which of the following graphs are isomorphic?



Solution-

Checking Necessary Conditions-

Condition-01:

- Number of vertices in graph G1 = 4
- Number of vertices in graph G2 = 4
- Number of vertices in graph G3 = 4

Here,

- All the graphs G1, G2 and G3 have same number of vertices.
- So, Condition-01 satisfies.

Condition-02:

- Number of edges in graph G1 = 5
- Number of edges in graph G2 = 5
- Number of edges in graph G3 = 4

Here,

- The graphs G1 and G2 have same number of edges.
- So, Condition-02 satisfies for the graphs G1 and G2.
- However, the graphs (G1, G2) and G3 have different number of edges.
- So, Condition-02 violates for the graphs (G1, G2) and G3.

Since Condition-02 violates for the graphs (G1, G2) and G3, so they can not be isomorphic.

∴ G3 is neither isomorphic to G1 nor G2.

Since Condition-02 satisfies for the graphs G1 and G2, so they may be isomorphic.

∴ G1 may be isomorphic to G2.

Now, let us continue to check for the graphs G1 and G2.

Condition-03:

- Degree Sequence of graph G1 = { 2 , 2 , 3 , 3 }
- Degree Sequence of graph G2 = { 2 , 2 , 3 , 3 }

Here,

- Both the graphs G1 and G2 have same degree sequence.
- So, Condition-03 satisfies.

Condition-04:

- Both the graphs contain two cycles each of length 3 formed by the vertices having degrees { 2 , 3 , 3 }
- It means both the graphs G1 and G2 have same cycles in them.
- So, Condition-04 satisfies.

Thus,

- All the 4 necessary conditions are satisfied.
- So, graphs G1 and G2 may be isomorphic.

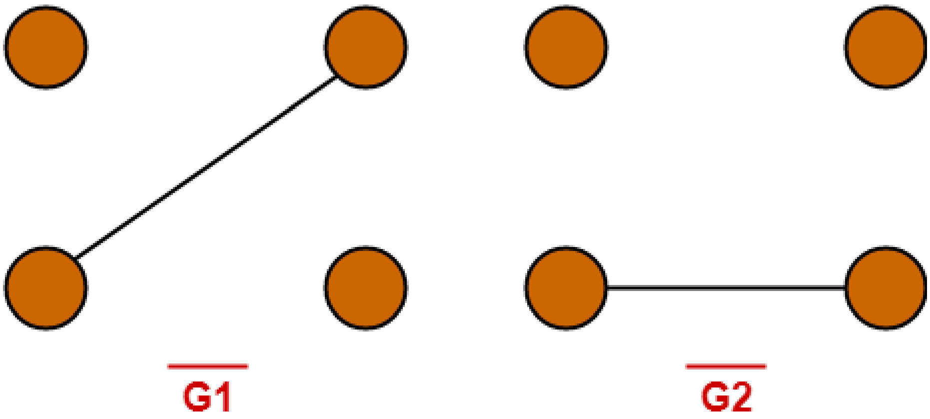
Now, let us check the sufficient condition.

Checking Sufficient Condition-

We know that two graphs are surely isomorphic if and only if their complement graphs are isomorphic.

So, let us draw the complement graphs of G1 and G2.

The complement graphs of G1 and G2 are-

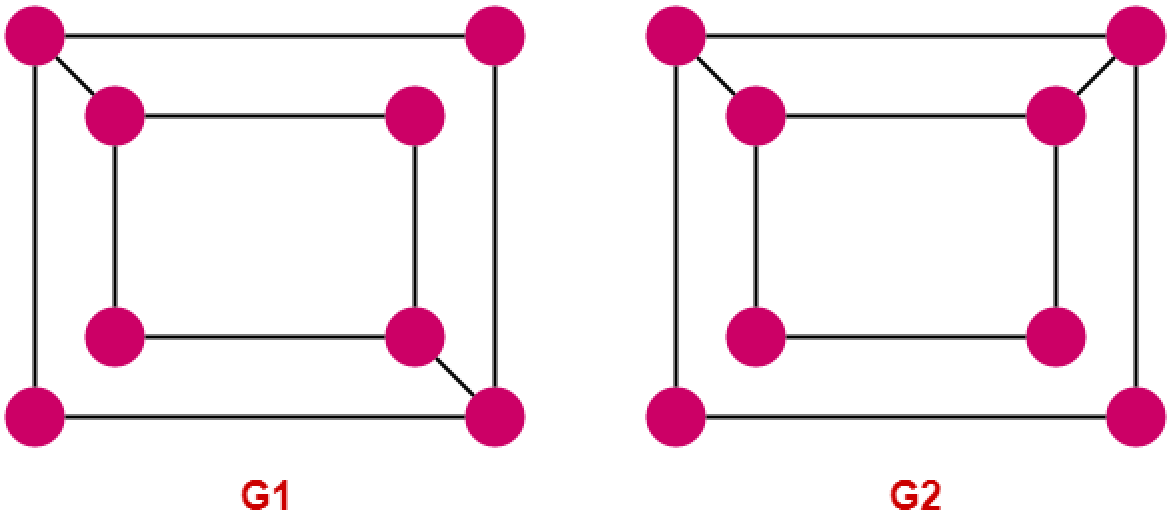


Clearly, Complement graphs of G1 and G2 are isomorphic.

∴ Graphs G1 and G2 are isomorphic graphs.

Problem-03:

Are the following two graphs isomorphic?



Solution-

Checking Necessary Conditions-

Condition-01:

- Number of vertices in graph G1 = 8
- Number of vertices in graph G2 = 8

Here,

- Both the graphs G1 and G2 have same number of vertices.
- So, Condition-01 satisfies.

Condition-02:

- Number of edges in graph G1 = 10
- Number of edges in graph G2 = 10

Here,

- Both the graphs G1 and G2 have same number of edges.
- So, Condition-02 satisfies.

Condition-03:

- Degree Sequence of graph G1 = { 2 , 2 , 2 , 2 , 3 , 3 , 3 , 3 }
- Degree Sequence of graph G2 = { 2 , 2 , 2 , 2 , 3 , 3 , 3 , 3 }

Here,

- Both the graphs G1 and G2 have same degree sequence.
- So, Condition-03 satisfies.

Condition-04:

- In graph G1, degree-3 vertices form a cycle of length 4.
- In graph G2, degree-3 vertices do not form a 4-cycle as the vertices are not adjacent.

Here,

- Both the graphs G1 and G2 do not contain same cycles in them.
- So, Condition-04 violates.

Since Condition-04 violates, so given graphs can not be isomorphic.

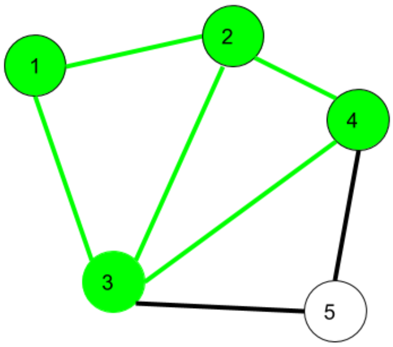
∴ G1 and G2 are not isomorphic graphs.

Walk, Path and Circuit

1. Walk –

A walk is a sequence of vertices and edges of a graph i.e. if we traverse a graph then we get a walk.

Note: Vertices and Edges can be repeated.



Here, 1->2->3->4->2->1->3 is a walk.

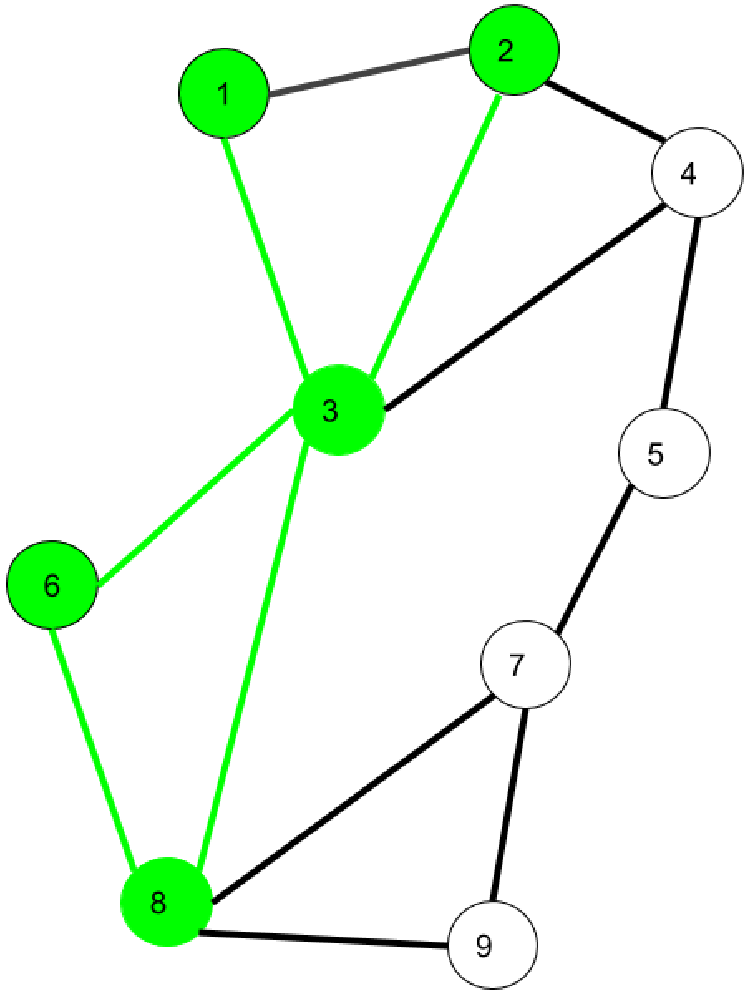
Walk can be open or closed.

Open walk- A walk is said to be an open walk if the starting and ending vertices are different i.e. the origin vertex and terminal vertex are different. **Closed walk-** A walk is said to be a closed walk if the starting and ending vertices are identical i.e. if a walk starts and ends at the same vertex, then it is said to be a closed walk.

In the above diagram:
1->2->3->4->5->3 is an open walk.
1->2->3->4->5->3->1 is a closed walk.

2. Trail –
Trail is an open walk in which no edge is repeated.

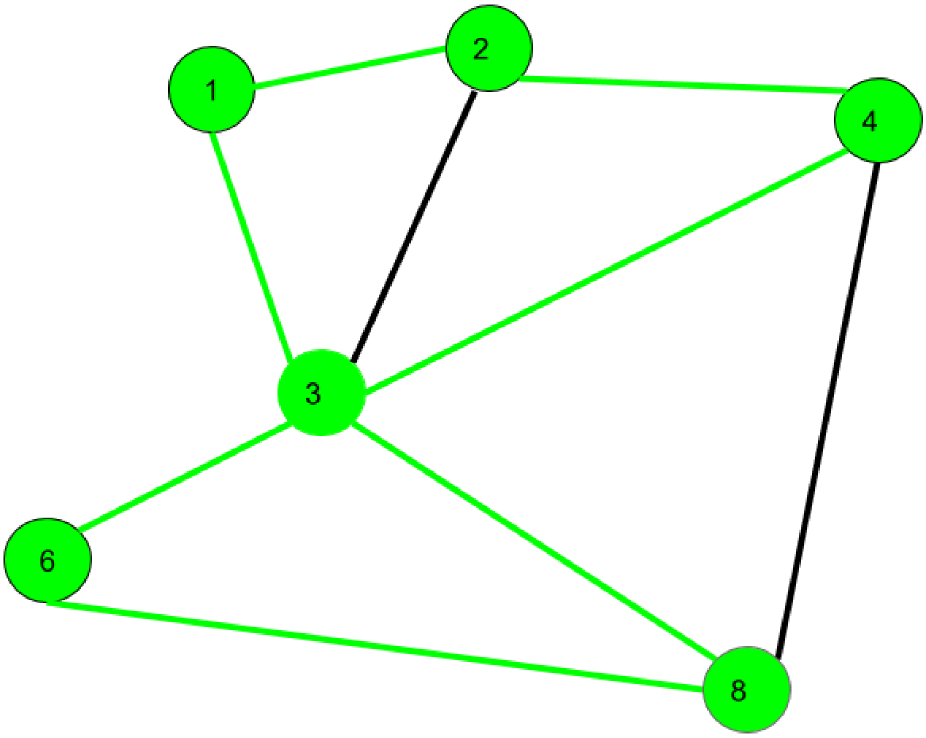
Vertex can be repeated.



Here 1->3->8->6->3->2 is trail
Also 1->3->8->6->3->2->1 will be a closed trail

3. Circuit –
Traversing a graph such that not an edge is repeated but vertex can be repeated and it is closed also i.e. it is a closed trail.

Vertex can be repeated.
Edge can not be repeated.



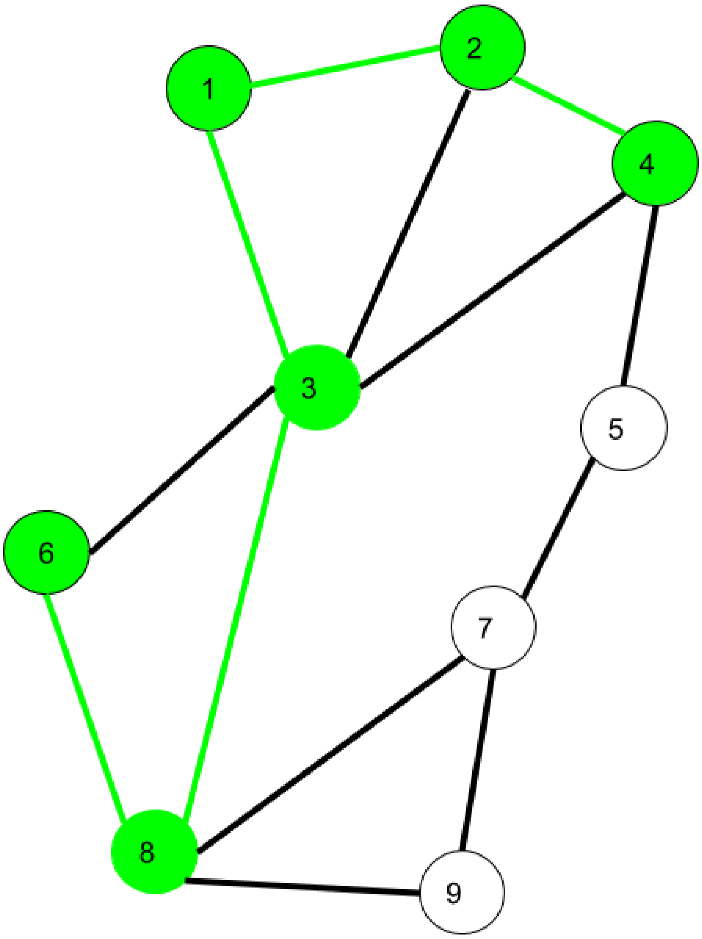
Here 1->2->4->3->6->8->3->1 is a circuit.

Circuit is a closed trail.

These can have repeated vertices only.

4. Path –
It is a trail in which neither vertices nor edges are repeated i.e. if we traverse a graph such that we do not repeat a vertex and nor we repeat an edge. As path is also a trail, thus it is also an open walk.

Vertex not repeated
Edge not repeated

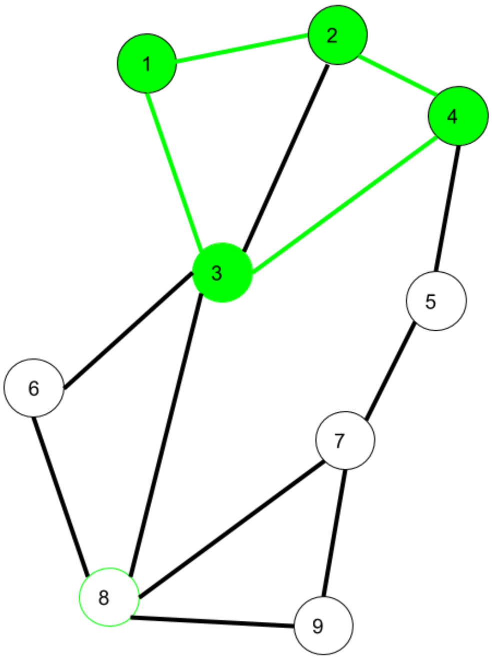


Here 6->8->3->1->2->4 is a Path

5. Cycle –

Traversing a graph such that we do not repeat a vertex nor we repeat a edge but the starting and ending vertex must be same i.e. we can repeat starting and ending vertex only then we get a cycle.

Vertex not repeated
Edge not repeated



Here 1->2->4->3->1 is a cycle.

Cycle is a closed path.

These can not have repeat anything (neither edges nor vertices).

Note that for closed sequences start and end vertices are the only ones that can repeat.

Binary Tree

If the outdegree of every node is less than or equal to 2, in a directed tree than the tree is called a binary tree. A tree consisting of the nodes (empty tree) is also a binary tree. A binary tree is shown in fig:

Basic Terminology:

Root: A binary tree has a unique node called the root of the tree.

Left Child: The node to the left of the root is called its left child.

Right Child: The node to the right of the root is called its right child

Parent: A node having a left child or right child or both are called the parent of the nodes.

Siblings: Two nodes having the same parent are called siblings.

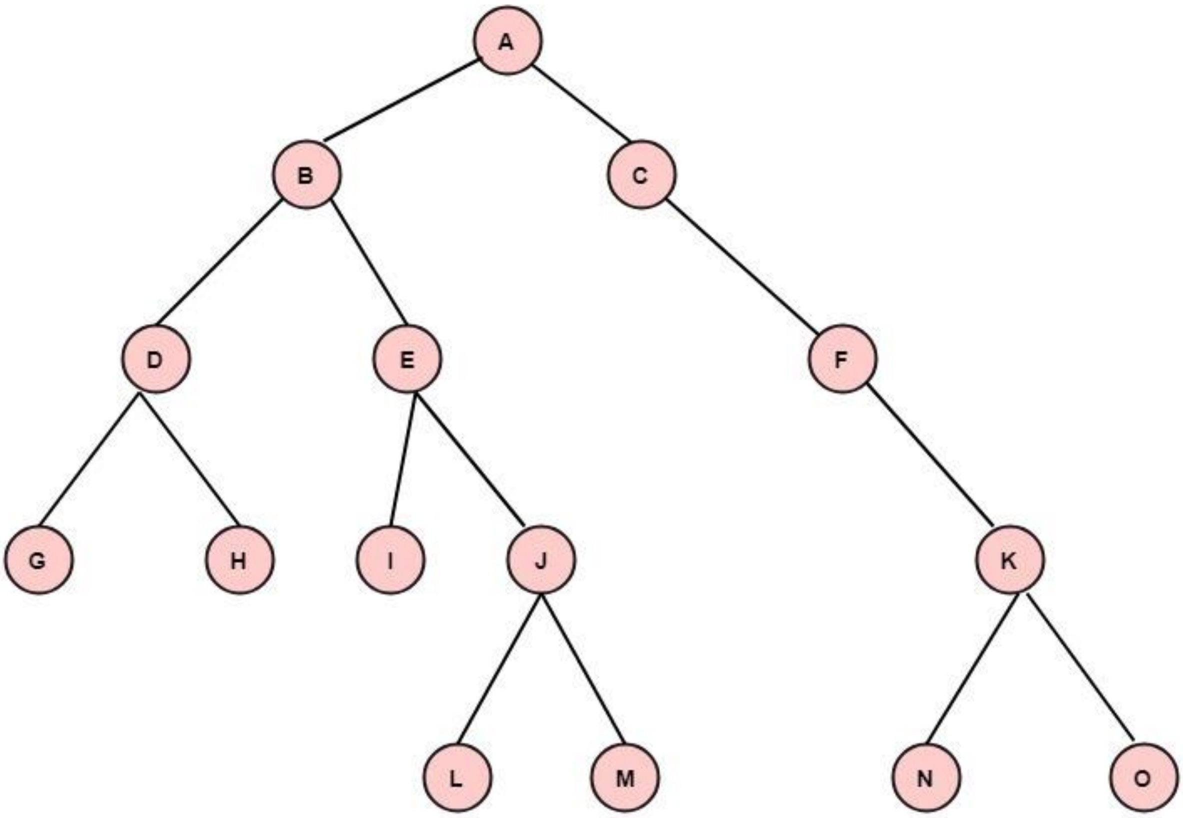
Leaf: A node with no children is called a leaf. The number of leaves in a binary tree can vary from one (minimum) to half the number of vertices (maximum) in a tree.

Descendant: A node is called descendant of another node if it is the child of the node or child of some other descendant of that node. All the nodes in the tree are descendants of the root.

Left Subtree: The subtree whose root is the left child of some node is called the left subtree of that node.

Example: For the tree as shown in fig:

- Which node is the root?
- Which nodes are leaves?
- Name the parent node of each node



Solution: (i) The node A is the root node.

(ii) The nodes G, H, I, L, M, N, O are leaves.

(iii)

Nodes	**Parent**
B, C	A
D, E	B
F	C
G, H	D
I, J	E
K	F
L, M	J
N, O	K

Right Subtree: The subtree whose root is the right child of some node is called the right subtree of that node

Level of a Node: The level of a node is its distance from the root. The level of root is defined as zero. The level of all other nodes is one more than its parent node. The maximum number of nodes at any level N is 2^N.

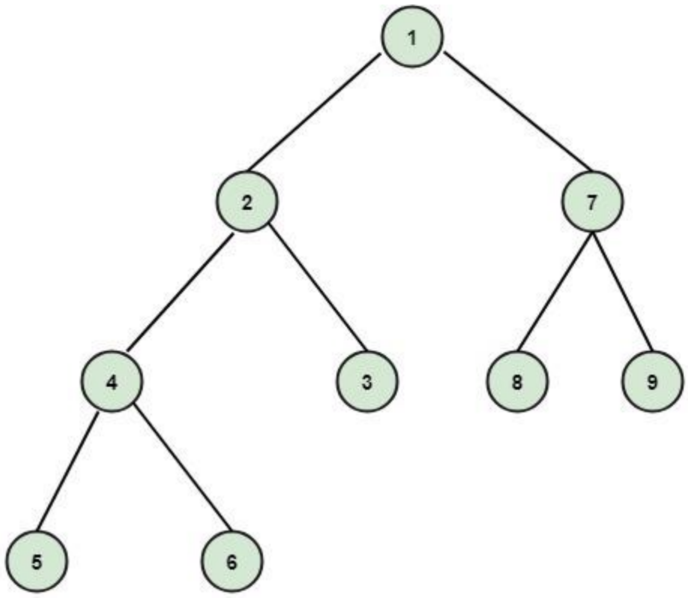
Depth or Height of a tree: The depth or height of a tree is defined as the maximum number of nodes in a branch of a tree. This is more than the maximum level of the tree, i.e., the depth of root is one. The maximum number of nodes in a binary tree of depth d is 2^d-1, where $d \geq 1$.

External Nodes: The nodes which have no children are called external nodes or terminal nodes.

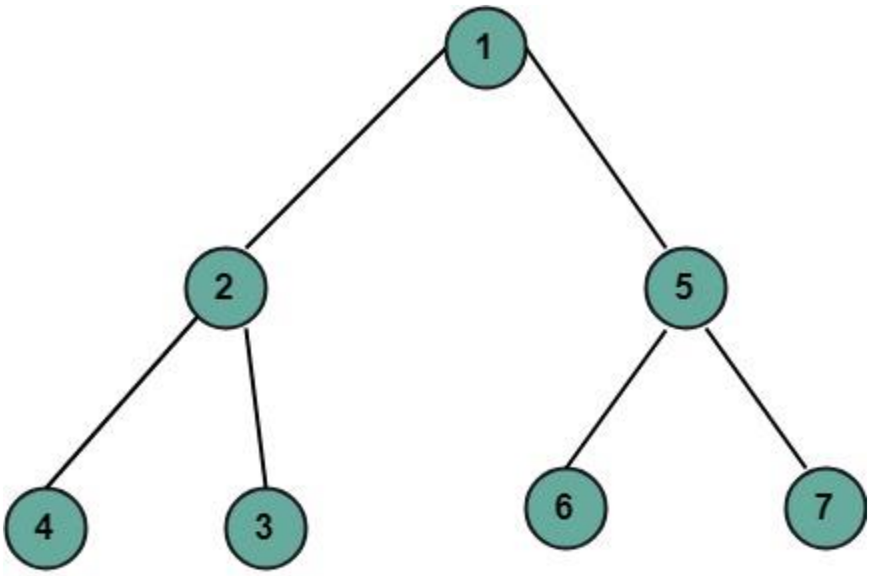
Internal Nodes: The nodes which have one or more than one children are called internal nodes or non-terminal nodes.

Complete Binary Tree: Complete binary tree is a binary tree if it is all levels, except possibly the last, have the maximum number of possible nodes as for left as possible. The depth of the complete binary tree having n nodes is $\log_2 n+1$.

Example: The tree shown in fig is a complete binary tree.



Full Binary Tree: Full binary tree is a binary tree in which all the leaves are on the same level and every non-leaf node has two children.



Full Binary Tree

Differentiate between General Tree and Binary Tree

General Tree	Binary Tree
1. There is no such tree having zero nodes or an empty general tree.	1. There may be an empty binary tree.

2. If some node has a child, then there is no such distinction.	2. If some node has a child, then it is distinguished as a left child or a right child.
3. The trees shown in fig are the same, when we consider them as general trees.	3. The trees shown in fig are distinct, when we consider them as binary trees, because in (4) is the right child of 2 while in (ii) 4 is a left child of 2.