SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF SCIENCE AND HUMANITIES

DEPARTMENT OF MATHEMATICS

UNIT – V – DISCRETE MATHEMATICS – SMTA 1302

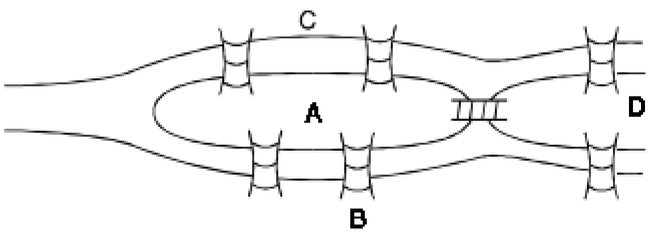
UNIT V - GRAPH THEORY

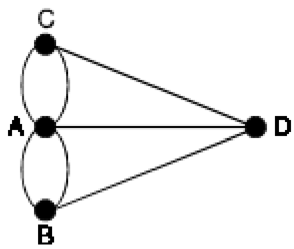
COURSE CONTENT: Introduction to graphs – Types of graphs (directed and undirected) – Basic terminology – Sub graphs – Representing graphs as incidence and adjacency matrix – Graph Isomorphism – Connectedness in Simple graphs, Paths and Cycles in graphs - Euler and Hamiltonian paths (statement only) – Tree – Binary tree (Definition and simple problems)

INTRODUCTION

The concept of graph theory is considered to have originated in 1736 with the publication of Euler's solution of the Konigsberg bridge problem. Euler (1707–1782) is regarded as the father of graph theory.

The Konigsberg Bridge Problem: The city of Konigsberg was located on the Pregel river in Prussia. The river divided the city into four separate landmasses, including the island of Kneiphopf. These four regions were linked by seven bridges as shown in the diagram. Residents of the city wondered if it were possible to leave home, cross each of the seven bridges exactly once, and return home. The Swiss mathematician Leonhard Euler thought about this problem and gave a solution.





The key to Euler's solution was in a very simple abstraction of the puzzle. Let us redraw our diagram of the city of Konigsberg by representing each of the land masses as a vertex and representing each bridge as an edge connecting the vertices corresponding to the land masses. We now have a graph that encodes the necessary information. The problem reduces to finding a "closed walk" in the graph which traverses each edge exactly once, this is called an Eulerian circuit. Euler proved such a circuit does not exist.

Graph theory is the study of points, lines and the ways in which sets of points can be connected by lines or arcs. Graphs in this context differ from the more familiar coordinate plots that portray mathematical relations and functions.

Graph theory has many colourful applications in many branches such as Physics, Chemistry, Communication Science, Computer technology, Electrical and Civil engineering, Architecture, Operations research, Genetics, Sociology, Economics etc.. It has proven useful in the design of integrated circuits (IC s) for computers and other electronic devices. These components more often called chips, contain complex, layered microcircuits that can be represented as sets of points interconnected by lines or arcs. Using graph theory, engineers develop chips with maximum component density and minimum total interconnecting conductor length. This is important for optimizing processing speed and electrical efficiency.

BASIC TERMINOLOGIES OF GRAPHS

A graph is usually denoted as G = (V, E), where V is called the **vertex set** of G and E is the **edge set** of G**.** The elements of the set V are called **vertices** or **points** or **nodes** and the members of the set E are called **edges** or lines or **arcs**.

The number of vertices in a graph G is called the **order of the graph** and is denoted by $|V|$. The number of edges in a graph is called the **size of the graph** and is denoted by $|E|$. A graph is **finite** if both its vertex set and edge set are finite. Otherwise it is an **infinite graph.** We study only finite graphs, so the term **graph** means only **finite graphs**.

A graph with p vertices and q edges is called a (p, q) graph. A graph with one vertex i.e., a (1, 0) graph is called **trivial graph** and all other graphs are non trivial. A graph with zero edges i.e., a (p, 0) graph is called **empty or null or void graph**. Each graph has a diagram associated with it. These diagrams are useful for understanding problems involving such graphs.

Adjacency

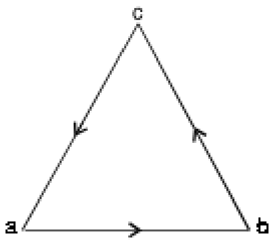
Two vertices v and w of a graph G are **adjacent** if there is an edge vw joining them, and the vertices v and w are then **incident** with such an edge. Similarly, two distinct edges e and *f* are adjacent if they have a vertex in common

DIRECTED AND UNDIRECTED GRAPHS

Directed graph

A directed graph G consists of a set V of vertices and a set E of edges such that $e \epsilon E$ is associated with an ordered pair of vertices. In other words, if each edge of the graph G has a direction then the graph is called **directed graph or digraph.**

In the diagram of directed graph, each edge is represented by an arrow or directed curve from initial point to the terminal point.



Suppose e = (a, b) is a directed edge in a digraph, then

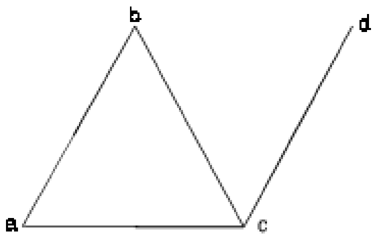
(i) a is called the initial vertex of e and b is the terminal vertex of e

(ii) e is said to be incident from vertex to vertex b.

Un-directed graph

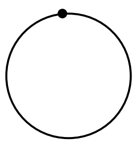
An un-directed graph G consists of set V of vertices and a set E of edges such that each edge e ∈ E is associated with an unordered pair of vertices. In other words, if each edge of the graph G has no direction then the graph is called un-directed graph.

Figure given below is an example of an undirected graph. An edge joining the vertex pair a and b can be referred as either (a, b) or (b, a).

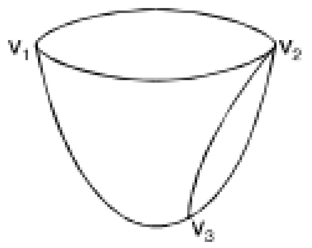


Loop : An edge of a graph that joins a vertex to itself is called loop.

Example:

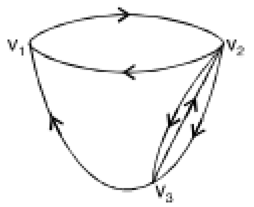


Multigraph: Two or more edges of a graph G joining the same pair of vertices are called multiple edges or parallel edges. The corresponding graph is called multigraph. In a multigraph no loops are allowed.



Un-directed multigraph

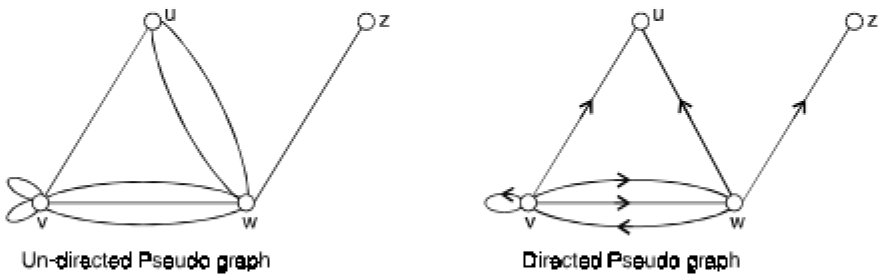
In the above figure there are two parallel edges joining nodes v_1, v_2 and v_2, v_3.



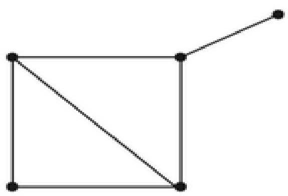
Directed multigraph

In the above figure there are two parallel edges associated with vertices v_2 and v_3

Pseudo graph: A graph, in which loops and multiple edges are allowed, is called a pseudo graph.



Simple graph: A graph with no loops and multiple edges is called a simple graph.

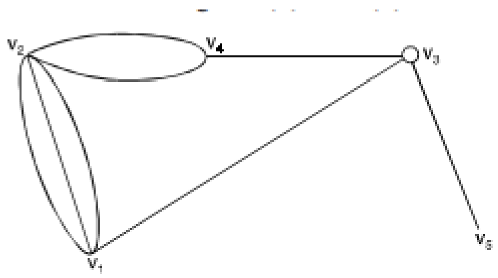


(a) Simple graph

DEGREE OF A VERTEX:

For an undirected graph, the number of edges incident on a vertex v_i with self-loops counted twice is called the degree of a vertex v_i and is denoted by deg (v_i) or deg v_i or $d(v_i)$. The degree of a vertex is also referred to as its **valency.**

For example let us consider the graph G given below. The degrees of vertices are deg $(v_1) = 4$, deg $(v_2) = 5$, deg $(v_3) = 5$, deg $(v_4) = 3$, and deg $(v_5) = 1$.



Isolated vertex: A vertex having no incident edge on it is called an isolated vertex. In other words vertex with zero degree is called an isolated vertex.

Pendent vertex or end vertex: A vertex of degree one, is called a pendent vertex or an end vertex and the corresponding edge is called the pendant edge. The vertex to which an end vertex is adjacent is called **support vertex**. In the above Figure, v_5 is a pendent vertex.

Degree Sequence: The vertex degrees of a graph arranged in non-increasing order is called degree sequence of the graph G. The degree sequence of the above graph is 5, 5, 4, 3, 1

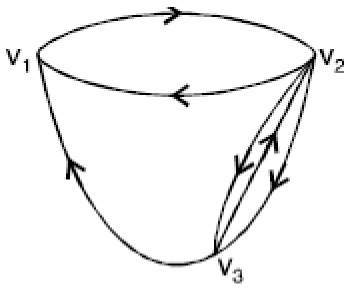
IN DEGREE and OUT DEGREE of a Vertex

In a digraph G, the number of edges beginning at vertex v_i is called the out degree of a vertex v_i, denoted by $deg_G^+(v_i)$ or out deg (v_i).

: In a digraph G, the number of edges ending at vertex v_i is called the in degree of a vertex v_i, denoted by $deg_G^-(v_i)$ or in deg (v_i).

A vertex with zero in degree is called a **source** and a vertex with zero out degree is called a **sink**.

The sum of the in degree and out degree of a vertex is called the total degree of the vertex.

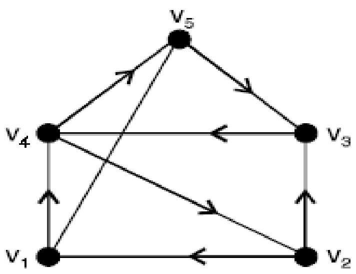


$$deg_G^-(v_1) = 2, deg_G^+(v_1) = 1, deg_G^-(2) = 2, deg_G^+(v_2) = 3, deg_G^-(v_3) = 2, deg_G^+(v_3) = 2$$

Note: For any directed graph the following property is true

$$\sum_{v \in V} deg^-(v) = \sum_{v \in V} deg^+(v) = |E|$$

Problem. *Find the in-degree and out-degree of each vertex of the following directed graph*



Solution.

in-degree $v1$ = 2, out-degree $v1$ = 1

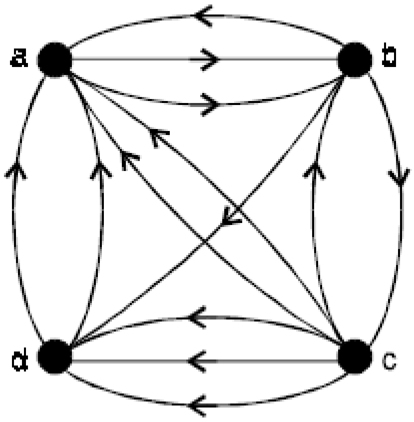
in-degree $v2$ = 2, out-degree $v2$ = 2

in-degree $v3$ = 2, out-degree $v3$ = 1

in-degree $v4$ = 2, out-degree $v4$ = 2

in-degree $v5$ = 0, out-degree $v5$ = 3

Problem. *Find the in-degree and out-degree of each vertex of the following directed graph*



Solution.

in-degree a = 6, out-degree a = 1
in-degree b = 1, out-degree b = 5
in-degree c = 2, out-degree c = 5
in-degree d = 2, out-degree d = 2.

Theorem 1: (THE HANDSHAKING THEOREM)

Statement: If G = (V, E) be an undirected graph with e edges, then $\sum_{v \epsilon V} deg_G(v) = 2e$. i.e., the sum of degrees of the vertices is an undirected graph is even.

(or)

If V = {v_1, v_2, ... v_n} is the vertex set and E is the edge set of a non directed graph G then $\sum_{i=1}^{n} deg_G(v_i) = 2|E|$

Proof :

Since the degree of a vertex is the number of edges incident with that vertex, the sum of the degree counts the total number of times an edge is incident with a vertex. Since every edge is incident with exactly two vertices, each edge gets counted twice, once at each end. Thus the sum of the degrees equals twice the number of edges.

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

Note : This theorem applies even if multiple edges and loops are present. The above theorem holds this rule that if several people shake hands, the total number of hands shaken must be even that is why the theorem is called handshaking theorem.

Corollary 1: In a non directed graph, the total number of odd degree vertices is even.

Proof :

Let G = (V, E) a non directed graph. Let U denote the set of even degree vertices in G and W denote the set of odd degree vertices.

Then $\sum_{v_i \epsilon V} deg_G(v_i) = \sum_{v_i \epsilon U} deg_G(v_i) + \sum_{v_i \epsilon W} deg_G(v_i)$

$\Rightarrow 2e - \sum_{v_i \epsilon U} deg_G(v_i) = \sum_{v_i \epsilon W} deg_G(v_i)$

$\Rightarrow \sum_{v_i \epsilon W} deg_G(v_i)$ is also even

∴ The number of odd vertices in G is even.

Theorem 2: If G is a directed graph, then $\sum_{i=1}^{n} deg_G^+(v_i) = \sum_{i=1}^{n} deg_G^-(v_i) = |E|$

Proof : Since when the degrees are summed, each edge contributes a count of one to the degree of each of the two vertices on which the edge is incident.

Corollary 2 : In any undirected graph there is an even number of vertices of odd degree.

Proof : Let W be the set of vertices of odd degree and let U be the set of vertices of even degree. Then $\sum_{v\epsilon U} deg_G(v) + \sum_{v\epsilon W} deg_G(v) = \sum_{v\epsilon V} deg_G(v) = 2|E|$

Certainly, $\sum_{v\epsilon U} deg_G(v)$ is even. Hence $\sum_{v\epsilon W} deg_G(v)$ is even.

$\Rightarrow |W|$ is even.

Corollary 3 : If k = δ(G) is the minimum degree of all the vertices of a non directed graph G, then

$$k|V| \le \sum_{v\epsilon V} deg_G(v) = 2|E|$$

In particular, if G is a k-regular graph, then

$$k|V| = \sum_{v\epsilon V} deg_G(v) = 2|E|$$

Problem. Show that the total number of odd degree vertices of a (p, q)-graph is always even. Solution. Let v_1, v_2 ...... v_k be the odd degree vertices in G.

Then, we have $\sum_{i=1}^{p} deg_G(v_i) = 2q$= even number

$\Rightarrow \sum_{i=1}^{k} deg_G(v_i) + \sum_{i=k+1}^{p} deg_G(v_i) =$ even number

$\Rightarrow \sum_{i=1}^{k} deg_G(v_i)$ = even number $- \sum_{i=k+1}^{p} deg_G(v_i)$

$\Rightarrow \sum_{i=1}^{k} deg_G(v_i)$ = even number – even number

= even number.

⇒ This implies that number of terms in the left-hand side of the equation is even.

Therefore, k is an even number.

Problem. Determine the number of edges in a graph with 6 vertices, 2 of degree 4 and 4 of degree 2.

Solution. Suppose the graph with 6 vertices has e number of edges. Therefore by Handshaking lemma. $\sum_{i=1}^{6} deg_G(v_i) = 2|e|$

$\Rightarrow d(v_1) + d(v_2) + d(v_3) + d(v_4) + d(v_5) + d(v_6) = 2e$

Now, given 2 vertices are of degree 4 and 4 vertices are of degree 2.

Hence the above equation becomes, (4 + 4) + (2 + 2 + 2 + 2) = 2e
⇒ 16 = 2e ⇒ e = 8.
Hence the number of edges in a graph with 6 vertices with given condition is 8.

Problem. How many vertices are needed to construct a graph with 6 edges in which each vertex is of degree 2?
Solution. Suppose these are n vertices in the graph with 6 edges. Also given the degree of each vertex is 2.
By handshaking lemma, $\sum_{i=1}^{n} deg_G(v_i) = 2|e|$= 2× 6 = 12
⇒ $d(v_1) + d(v_2) + \ldots\ldots + d(v_n) = 12$
⇒ $\underbrace{2 + 2 + \cdots + 2}_{n\ times} = 12$
⇒ 2n = 12
⇒ n = 6 vertices are needed.

Problem. It is possible to draw a simple graph with 4 vertices and 7 edges ? Justify.
Solution. In a simple graph with n-vertices, the maximum number of edges will be $\frac{n(n-1)}{2}$.
Hence a simple graph with 4 vertices will have at most $\frac{4\times\times 3}{2} = 6$edges.
Therefore, a simple graph with 4 vertices cannot have 7 edges.
Hence such a graph does not exist.

Problem. Show that there exists no simple graph corresponds to the following degree sequence : (i) 0, 2, 2, 3, 4 (ii) 1, 1, 2, 3 (iii) 2, 2, 3, 4, 5, 5 (iv) 2, 2, 4, 6.
Solution. (i) to (iii) : There are odd number of odd degree vertices in the graph. Hence there exists no graph corresponds to this degree sequence.
(iv) Number of vertices in the graph is four and the maximum degree of a vertex is 6, which is not possible as the maximum degree cannot exceed one less than the number of vertices.

Problem. Show that the following sequence 6, 5, 5, 4, 3, 3, 2, 2, 2 is graphical.
Solution.
We can reduce the sequence as follows :
Given sequence 6, 5, 5, 4, 3, 3, 2, 2, 2
Reducing first 6 terms by 1 counting from second term 4, 4, 3, 2, 2, 1, 2, 2.
Writing in decreasing order 4, 4, 3, 2, 2, 2, 2, 1
Reducing first 4 terms by 1 counting from second 3, 2, 1, 1, 2, 2, 1
Writing in decending order 3, 2, 2, 2, 1, 1, 1
Reducing first 3 terms by 1, counting from second 1, 1, 1, 1, 1, 1

Sequence 1, 1, 1, 1, 1, 1 is graphical.
Hence the given sequence is also graphical

Problem. Show that the sequence 6, 6, 6, 6, 4, 3, 3, 0 is not graphical.
Solution. To prove that the sequence is not graphical.
The given sequence is 6, 6, 6, 6, 4, 3, 3, 0
Resulting the sequence 5, 5, 5, 3, 2, 2, 0
Again consider the sequence 4, 4, 2, 1, 1, 0
Repeating the same 3, 1, 0, 0, 0
Since there exists no simple graph having one vertex of degree three and other vertex of degree one. The last sequence is not graphical.
Hence the given sequence is also not graphical.

Problem. Show that the maximum number of edges in a simple graph with n vertices is $\frac{n(n-1)}{2}$. . Solution. By the handshaking theorem,

$\sum_{i=1}^{n} deg_G(v_i) = 2|e|$where e is the number of edges with n vertices in the graph G.

$\Rightarrow d(v_1) + d(v_2) + ... + d(v_n) = 2e$ .... (1)

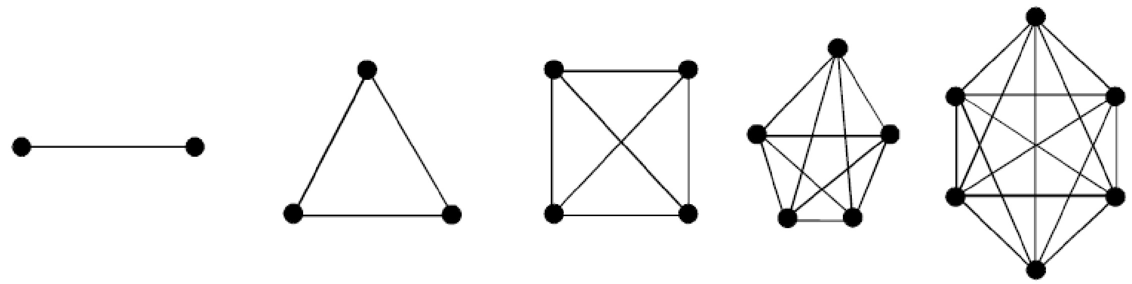
We know that the maximum degree of each vertex in the graph G can be (n – 1).
Therefore, equation (1) reduces $\underbrace{(n-1) + (n-1) + \cdots + (n-1)}_{n\ times} = 2e$

$\Rightarrow n(n – 1) = 2e$
$\Rightarrow e = \frac{n(n-1)}{2}$.
Hence the maximum number of edges in any simple graph with n vertices is $\frac{n(n-1)}{2}$.

SOME SPECIAL GRAPHS:

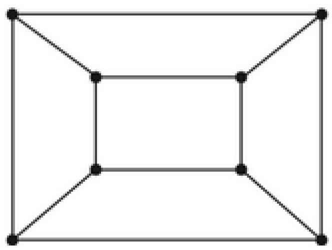
COMPLETE GRAPH

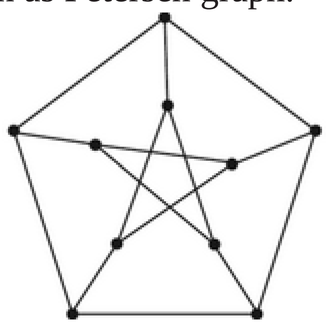
A simple graph G is said to be **complete** if every vertex in G is connected with every other vertex.
i.e., if G contains exactly one edge between each pair of distinct vertices.
A complete graph is usually denoted by **K*n*.** It should be noted that K*n* has exactly $\frac{n(n-1)}{2}$edges.
The figure given below shows complete graphs K_1 to K_6



REGULAR GRAPH

A graph in which all vertices are of **equal degree,** is called a **regular graph.**
If the degree of each vertex is *r*, then the graph is called a regular **graph of degree *r*.**
Note 1: Every null graph is regular of degree zero.
Note 2: The complete graph K*n* is a regular of degree *n* – 1.
Note 3: If G has *n* vertices and is regular of degree *r*, then G has edges.
Note 4: The figure given below shows 3 regular graphs which are also called as cubic graphs. The socond graph is also known as Petersen graph.





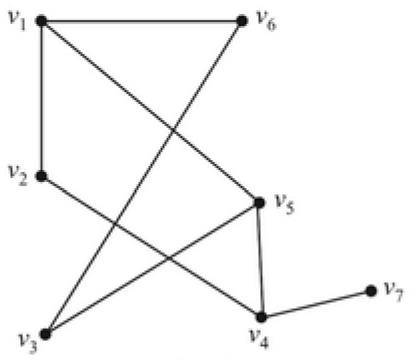
BIPARTITE GRAPH

A graph G is said to be **bipartite** if its vertex set can be partitioned into two subsets such that no two vertices in the same partition are adjacent. In other words if the simple graph G(V, E) can be partitioned into two subsets V_1 and V_2 such that every edge of G connects a vertex in V_1 to a vertex in V_2 and no edge in G connects either two vertices in V_1 or V_2 then G is called a **bipartite graph.**
If each vertex of V_1 is connected with every vertex of V_2 by an edge, Then G is said to be a **complete bipartite graph.** If V_1 contains m vertices and V_2 contains n vertices then the complete bipartite graph is denoted by $K_{m, n}$.

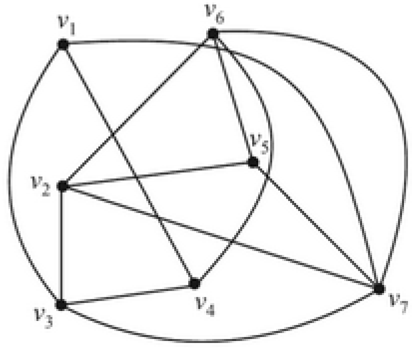
The following figure shows bipartite and complete bipartite graph

THE COMPLEMENT OF A GRAPH

Let G be a simple graph. The complement of G denoted by G^c has the same vertex set as G and two vertices in G and G^c are adjacent if and only if they are not adjacent in G.
The graph G and its complement G^c are depicted below



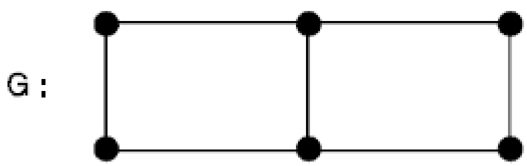
(a) G

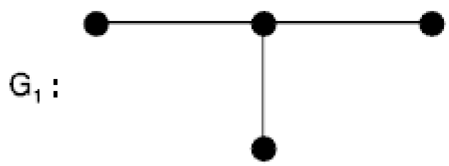


(b) G^c

SUBGRAPH

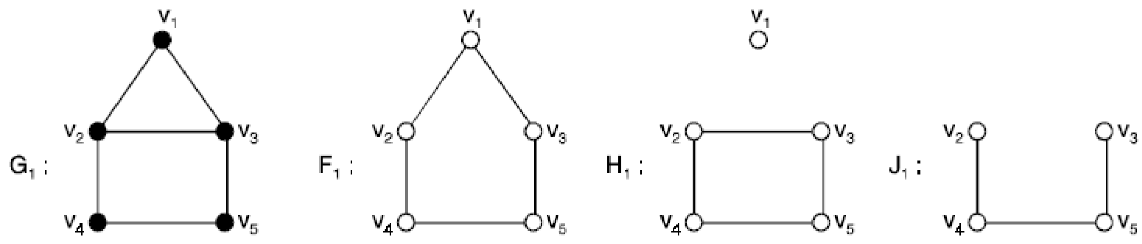
If G and H are two graphs with vertex sets V(H), V(G) and edge sets E(H) and E(G) respectively such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ then we call H as a subgraph of G or G as a supergraph of H.
In the figure given below G_1 is a subgraph of graph G.





SPANNING SUBGRAPH

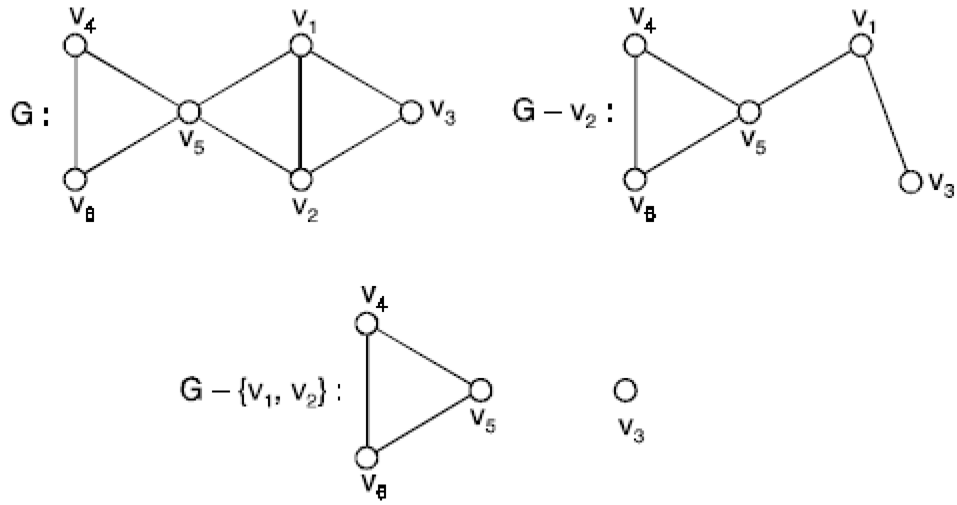
A graph H is called a subgraph of a graph G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G$.
If $V(H) \subset V(G)$ and $E(H) \subset E(G)$ then H is called a **proper subgraph** of G.
If V(H) = V(G) then we say that H is a **spanning subgraph** of G.
A spanning subgraph need not contain all the edges in G. The graphs F1 and H1 of the figure shown below are spanning subgraphs of G1, but J1 is not a spanning subgraph of G1.



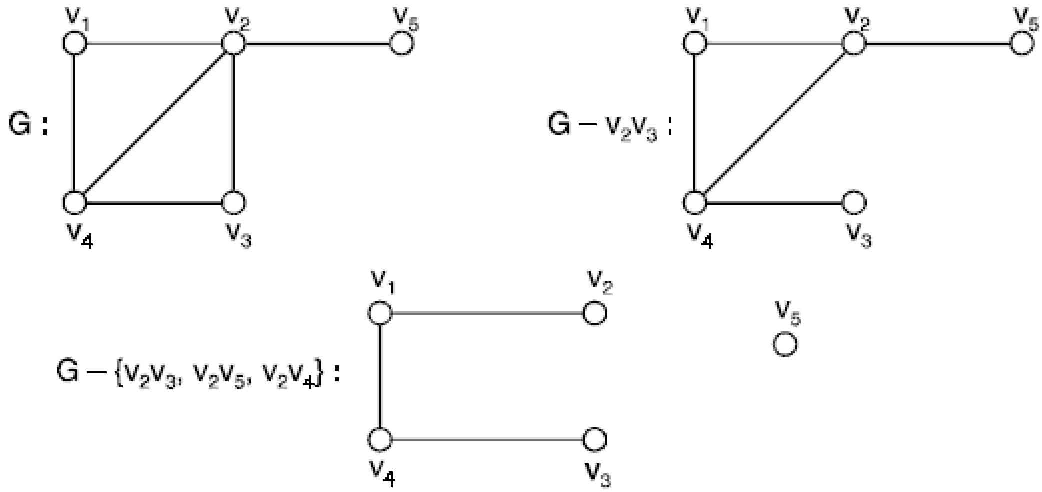
Removal of a vertex and an edge

The removal of a vertex *vi* from a graph G result in that subgraph G – *vi* of G containing of all vertices in G except *vi* and all edges not incident with *vi*. Thus G – *vi* is the maximal subgraph of G not containing *vi*. On the otherhand, the removal of an edge *xj* from G yields the spanning subgraph G – *xj* containing all edges of G except *xj*. Thus G – *xj* is the maximal subgraph of G not containing edge *xj*.

The following figure shows deletion of vertices and deletion of edges from a graph



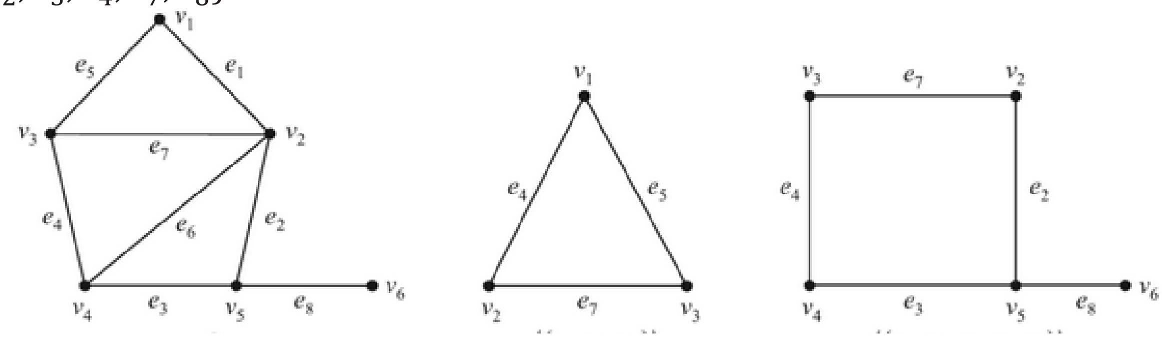
The following figure shows deletion of edges from a graph



INDUCED SUB GRAPH:

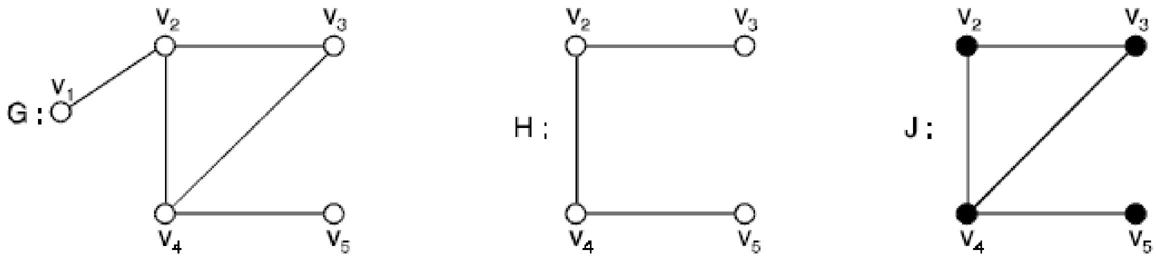
Let G be a graph with vertex set V(G), edge set E(G) and S be a non empty subset of V(G). A subgraph of G whose vertex set is S and all edges of G which have both their ends in S is known as the subgraph induced by S and is denoted by G[S] or $< S >$. Any subgraph induced by a set of vertices will be called a **vertex induced subgraph or simply an induced sub graph**. In other words a sub graph H of a graph G where V(H) $\subseteq$ V(G) and E(H) consists of only thoe edges that are incident on the elements of V(H), is called an **induced sub graph** of G.

Let M be a non empty subset of E(G). A subgraph of G whose edge set is M and whose vertices are the ends of edges in M, is said to be a subgraph induced by M and is denoted by G[M] or $< M >$. The second figure below displays the vertex induced sub graph of graph G induced by vertex set $\{v_1, v_2, v_3\}$ and the third image in the figure shown below is the edge induced sub graph of G induced by the edge set $\{ e_2, e_3, e_4, e_7, e_8\}$

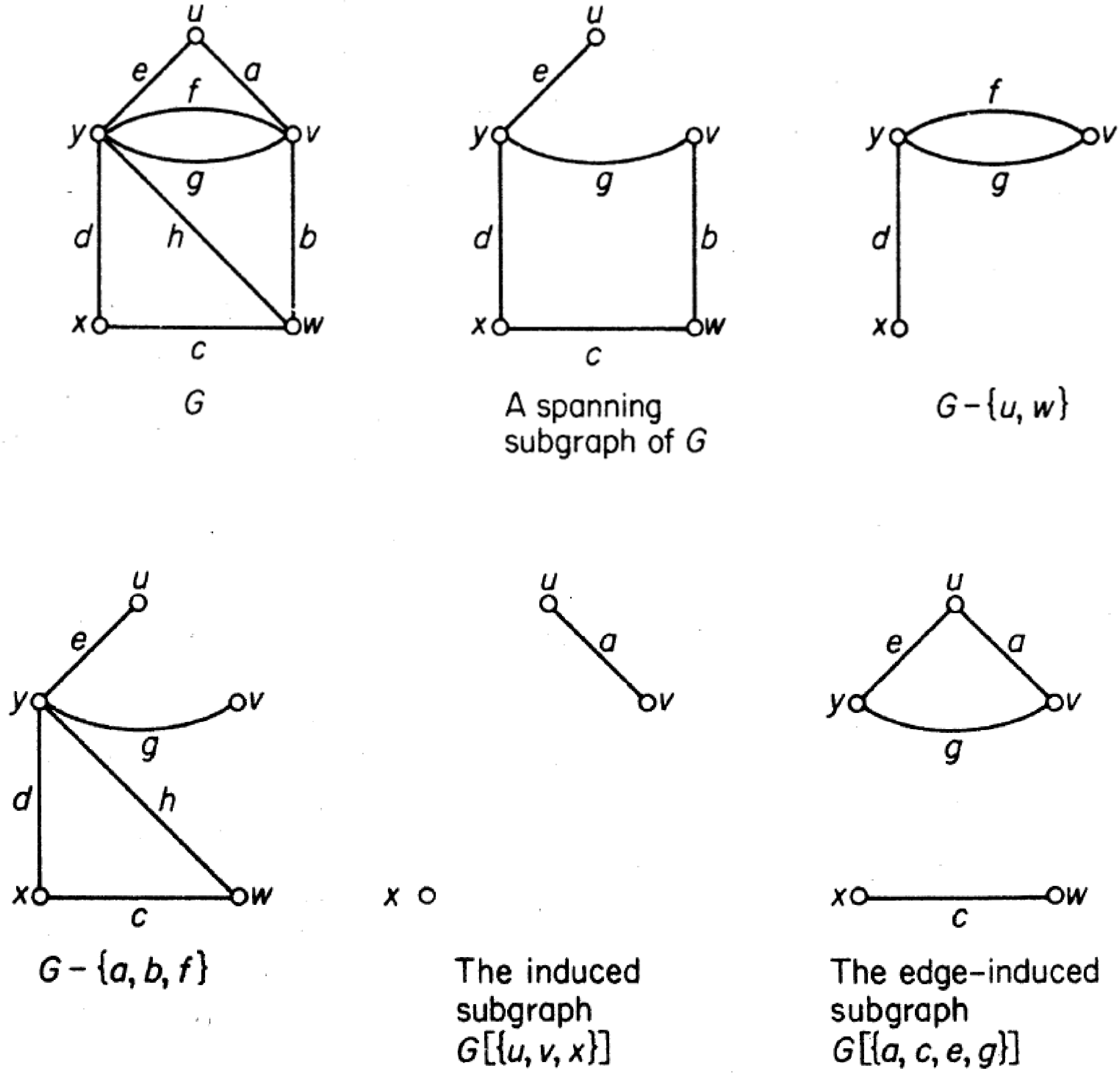


Graph G Induced sub graphs of graph G

Example H is not an induced subgraph since $v4v1 \in E(G)$, but $v4v3 \notin E(H)$.



Example for spanning sub graph, vertex induced sub graph and edge induced sub graph



G

A spanning subgraph of G

$G - \{u, w\}$

$G - \{a, b, f\}$

The induced subgraph $G[\{u, v, x\}]$

The edge-induced subgraph $G[\{a, c, e, g\}]$

GRAPHS ISOMORPHISM

Let G1 = (V1, E1) and G2 = (V2, E2) be two graphs. A function $f : V1 \rightarrow V2$ is called a graphs isomorphism if

(*i*) f is one-to-one and onto.

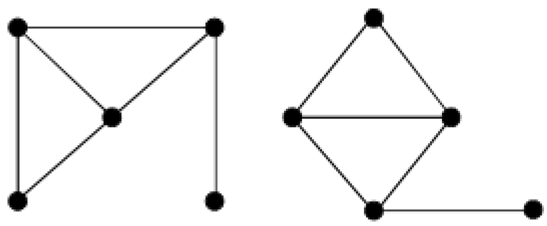
(*ii*) for all $a, b \in V1$, $\{a, b\} \in E1$ if and only if $\{f(a), f(b)\} \in E2$ when such a function exists, G1 and G2 are called isomorphic graphs and is written as $G1 \cong G2$. In other words, two graphs G1 and G2 are said to be isomorphic to each other if there is a one to- one correspondence between their vertices and between edges such that incidence relationship is preserved. It is written as $G1 \cong G2$ or G1 = G2.

The necessary conditions for two graphs to be isomorphic are

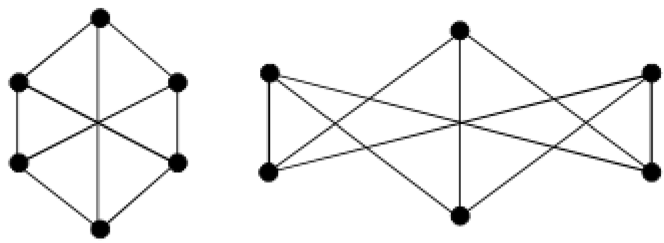
1. Both must have the **same number of vertices**
2. Both must have the **same number of edges**
3. Both must have **equal number of vertices** with the **same degree.**
4. They must have the same degree sequence and same cycle vector ($c1$, ..., cn), where ci is
the number of cycles of length i.

The isomorphic pair of graphs are shown below

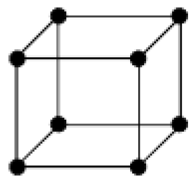
Example 1:

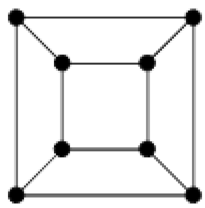


Example 2:

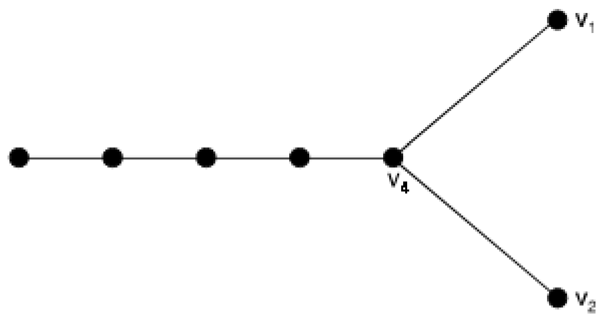


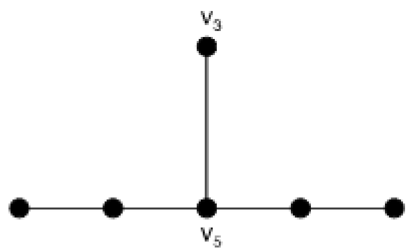
Example 3:



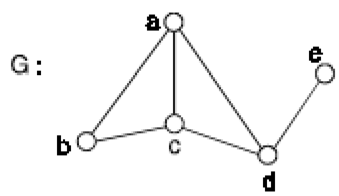


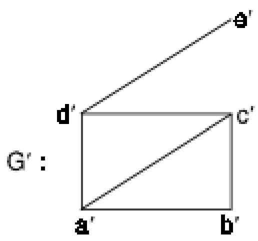
Example of two graphs that are not isomorphic





Problem. *Show that the following graphs are isomorphic*

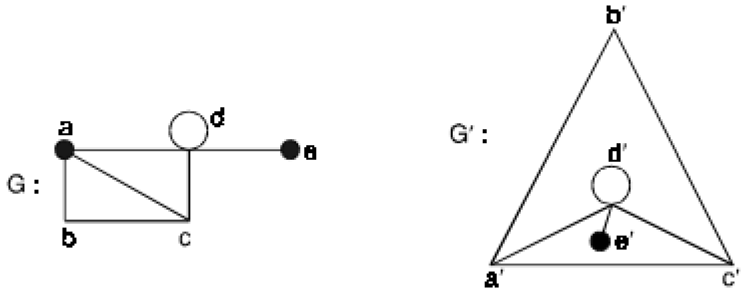




Solution. Let f : G → G′ be any function defined between two graphs degrees of the graph G and
G′ are as follows :
deg (G) deg (G′)
deg (a) = 3 deg (a') = 3
deg (b) = 2 deg (b') = 2
deg (c) = 3 deg (c') = 3
deg (d) = 3 deg (d') = 3
deg (e) = 1 deg (e') = 1
Each has 5-vertices and 6-edges.
$d(a) = d(a') = 3$
$d(b) = d(b') = 2$
$d(c) = d(c') = 3$
$d(d) = d(d') = 3$

$d(e) = d(e') = 1$
Hence the correspondence is $a - a'$, $b - b'$, ..., $e - e'$.
Therefore, the given two graphs are isomorphic.

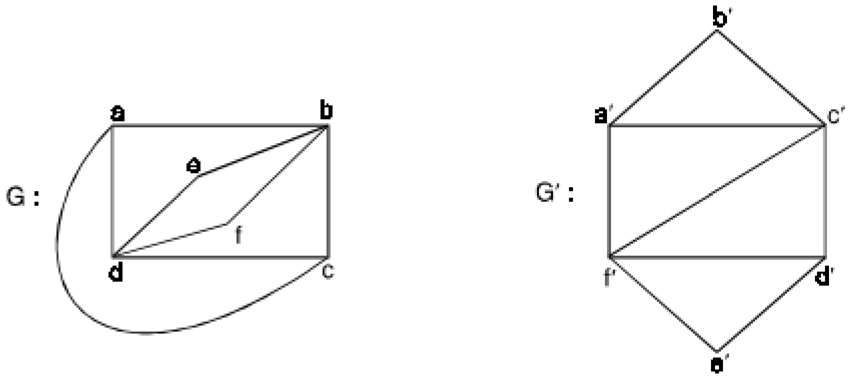
Problem. *Show that the following graphs are isomorphic.*



Solution. Let $f : G \to G'$ be any function defined between two graphs degrees of the graphs G
and G′ are as follows :
deg (G) deg (G′)
deg $(a) = 3$ deg $(a') = 3$
deg $(b) = 2$ deg $(b') = 2$
deg $(c) = 3$ deg $(c') = 3$

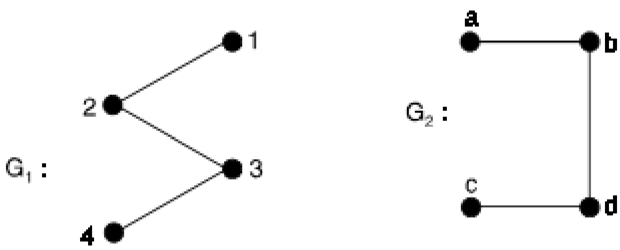
deg $(d) = 5$ deg $(d') = 5$
deg $(e) = 1$ deg $(e') = 1$
Each has 5-vertices, 6-edges and 1-circuit.
$\deg(a) = \deg(a') = 3$
$\deg(b) = \deg(b') = 2$
$\deg(c) = \deg(c') = 3$
$\deg(d) = \deg(d') = 5$
$\deg(e) = \deg(e') = 1$
Hence the correspondence is $a - a'$, $b - b'$, ..., $e - e'$.
Therefore, the given two graphs G and G′ are isomorphic.

Problem. *Are the 2-graphs, is given below, is isomorphic ? Give a reason.*

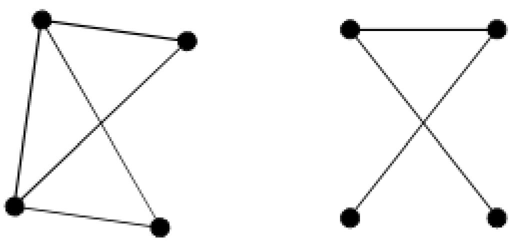


Solution. Let us enumerate the degree of the vertices
Vertices of degree 4 : $b - f'$
$d - c'$
Vertices of degree 3 : $a - a'$
$c - d'$
Vertices of degree 2 : $e - b'$
$f - e'$
Now the vertices of degree 3, in G are a and c and they are adjacent in G′, while these are a' and
d' which are not adjacent in G′.
Hence the 2-graphs are not isomorphic.

Problem. *For each pair of graphs shown, either label the graphs so as to exhibit an isomorphism or explain why the graphs are not isomorphic.*

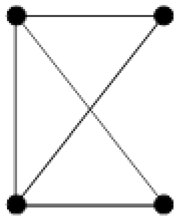


(*i*)

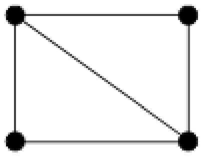


(a) (b)

(ii)

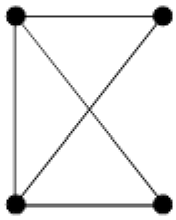


(a)

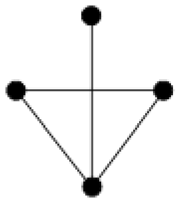


(b)

(iii)

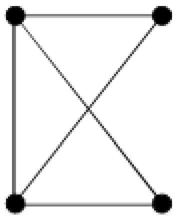


(a)

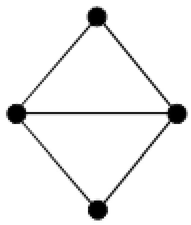


(b)

(iv)



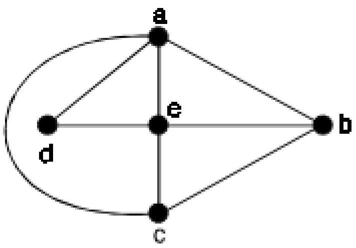
(a)

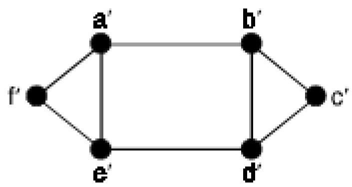


(b)

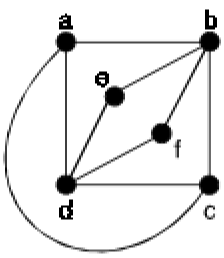
Problems. *Are the 2-graphs, is given below, is isomorphic ? Give a reason.*

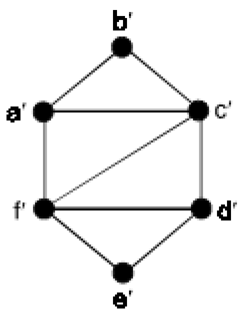
(i)



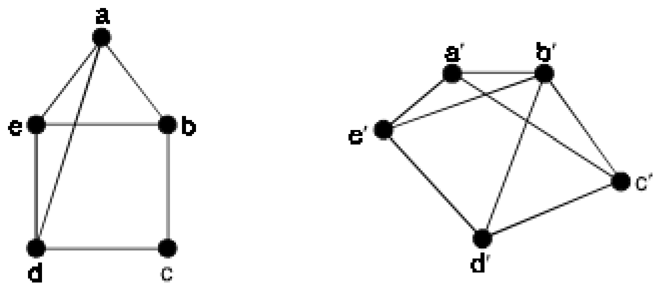


(ii)





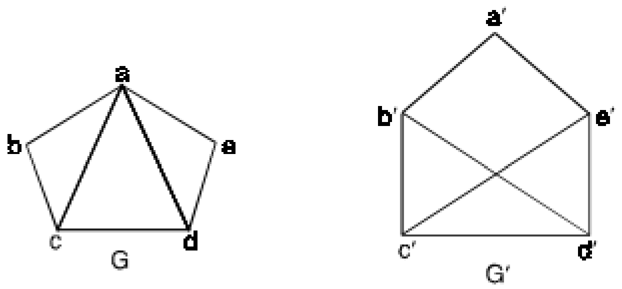
(iii)



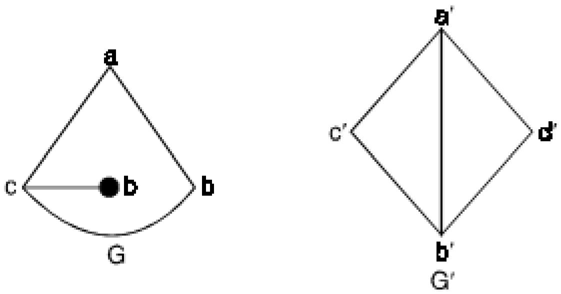
\

Problem. *Find whether the following pairs of graphs are isomorphic or not*

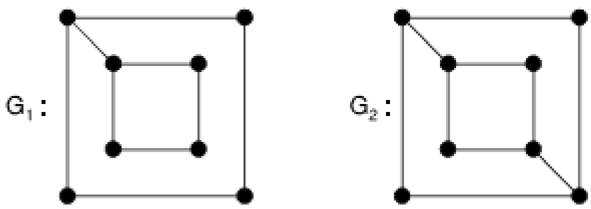
(i)

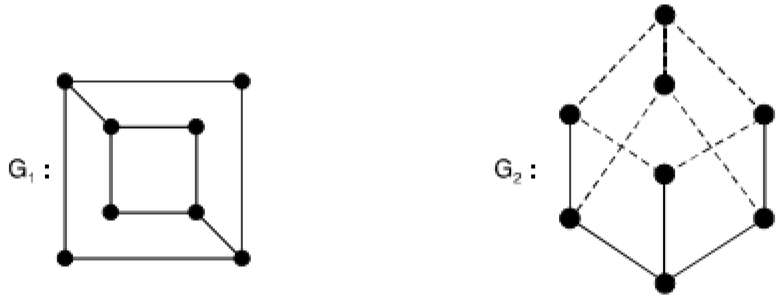


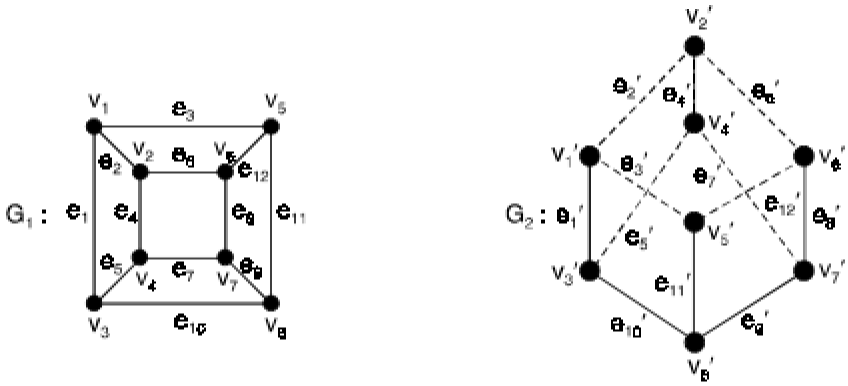
(ii)

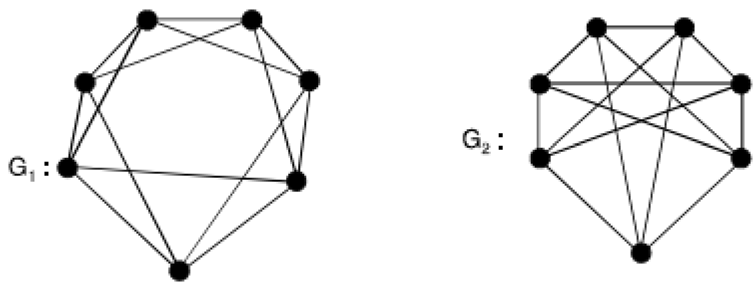


Problem. *Consider two graphs G1 and G2 as shown below, show that the graphs G1 and*
G2 are isomorphic.









REPRESENTATION OF GRAPHS

Although a diagrammatic representation of a graph is very convenient for a visual study but this
is only possible when the number of nodes and edges is reasonably small.Two types of representation are given below :

Matrix representation

The matrix are commonly used to represent graphs for computer processing. The advantages of representing the graph in matrix form lies on the fact that many results of matrix algebra can be readily
applied to study the structural properties of graphs from an algebraic point of view. There are number of

matrices which one can associate witch any graph. We shall discuss adjacency matrix and the incidence
matrix.

ADJACENCY MATRIX

Representation of undirected graph

The adjacency matrix of a graph G with *n* vertices and no parallel edges is an *n* by *n* matrix A = {*aij*}
whose elements are given by aij =

$$\begin{cases} 1 & \text{if there is an edge between } i\text{th and } j\text{th vertices} \\ 0 & \text{if there is an edge between } i\text{th and } j\text{th vertices} \end{cases}$$

Note that for a given graph, the adjacency matrix is based on ordering chosen for the vertices. Hence, there are as many as *n* ! different adjacency matrices for a graph with *n* vertices, since there are *n* ! different ordering of *n* vertices. However, any two such adjacency matrices are closely related in that one can be obtained from
the other by simply interchanging rows and columns.
There are a number of observations that one can make about the adjacency matrix A of a graph G. They are
(*i*) A is symmetric *i.e. aij* = *aji* for all *i* and *j*
(*ii*) The entries along the principal diagonal of A all zeros if and only if the graph has no self loops. A self loop at the vertex corresponding to *aij* = 1.
(*iii*) If the graph is simple (no self loop, no parallel edges), the degree of vertex equals the number of 1's in the corresponding row or column of A.
(*iv*) The (*i, j*) entry of A*m* is the number of paths of length (no. of occurrence of edges) *m* from vertex *vi* to vertex *vj*.
(*v*) If G be a graph with *n* vertices *v*1, *v*2, ... *vn* and let A denote the adjacency matrix of G with respect to this listing of the vertices. Let B be the matrix and B = A + A2 + A3 + ... + A*n* – 1
Then G is a connected graph if B has no zero entries of the main diagonal.
This result can be also used to check the connectedness of a graph by using its adjacency matrix.

Adjacency can also be used to represent undirected graphs with loops and multiple edges. A loop at the vertex *v*1 is represented by a 1 at the (*i, j*)th position of the adjacency matrix. When multiple edges are present, the adjacency matrix is no longer a zero-one matrix, since the (*i, j*)th entry equals the number of edges these are associated to {*vi* – *vj*}.

All undirected graphs, including multigraphs and pseudographs, have symmetric adjacency matrices.

Representation of directed graph

The adjacency matrix of a diagonal D, with *n* vertices is the matrix A = $\{aij\}n \times n$ in which

$$aij = \begin{cases} 1 & \text{if arc } \{vi - vj\} \text{ is in D} \\ 0 & \text{otherwise} \end{cases}$$

One can make a number of observations about the adjacency matrix of a diagonal.

Observations

(*i*) A is not necessary symmetric, since there may not be an edges from *vi* to *vj* when there is an edge from *vi* to *vj*.

(*ii*) The sum of any column of *j* of A is equal to the number of arcs directed towards *vj*

(*iii*) The sum of entries in row *i* is equal to the number of arcs directed away from vertex *vi* (out degree of vertex *vi*)

(*iv*) The (*i*, *j*) entry of A*m* is equal to the number of path of length *m* from vertex *vi* to vertex *vj* entries of AT. A shows the in degree of the vertices.

The adjacency matrices can also be used to represent directed multigraphs. Again such matrices are not zero-one matrices when there are multiple edges in the same direction connecting two vertices.

In the adjacency matrix for a directed multigraph *aij* equals the number of edges that are associated to (*vi*, *vj*).

INCIDENCE MATRIX

Representation of undirected graph

Consider a undirected graph G = (V, E) which has *n* vertices and *m* edges all labelled. The

incidence matrix B = $\{bij\}$, is then $n \times m$ matrix,

where $b_{ij} = \begin{cases} 1 & when\ edge\ e_j\ is\ incident\ with\ vertex\ v_i \\ 0 & otherwise \end{cases}$

We can make a number of observations about the incidence matrix B of G.

(*i*) Each column of B comprises exactly two unit entries.
(*ii*) A row with all 0 entries corresponds to an isolated vertex.
(*iii*) A row with a single unit entry corresponds to a pendent vertex.
(*iv*) The number of unit entries in row i of B is equal to the degree of the corresponding vertex v_i.
(*v*) The permutation of any two rows (any two columns) of B corresponds to a labelling of the vertices (edges) of G.
(*vi*) Two graphs are isomorphic if and only if their corresponding incidence matrices differ only by a permutation of rows and columns.
(*vii*) If G is connected with n vertices then the rank of B is $n - 1$.

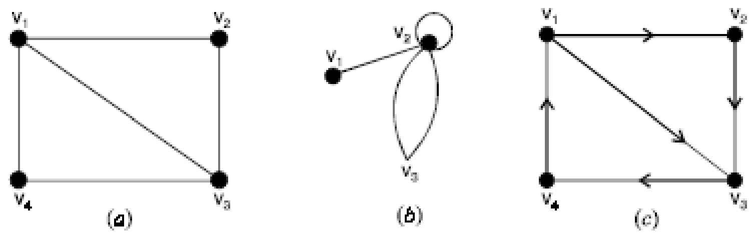
Incidence matrices can also be used to represent multiple edges and loops. Multiple edges are represented in the incidence matrix using columns with identical entries. Since these edges are incident with the same pair of vertices. Loops are represented using a column with exactly one entry equal to 1, corresponding to the vertex that is incident with this loop.

Representation of directed graph

The incidence matrix B = $\{b_{ij}\}$ of digraph D with n vertices and m edges is the $n \times m$ matrix in which

$$b_{ij} = \begin{cases} 1 & if\ arc\ j\ is\ directed\ away\ from\ vertex\ v_i \\ -1 & if\ arc\ j\ is\ directed\ towards\ vertex\ v_i \\ 0 & otherwise \end{cases}$$

Problem 14. *Use adjacency matrix to represent the graphs shown in Figure below*



Solution. We order the vertices in Figure (*a*) as v_1, v_2, v_3 and v_4.
Since there are four vertices, the adjacency matrix representing the graph will be a square matrix of order four. The required adjacency matrix A is

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

We order the vertices in Figure (*b*) as v_1, v_2 and v_3. The adjacency matrix representing the graph is given by

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 2 \\ 0 & 2 & 0 \end{bmatrix}$$

Taking the order of the vertices in Figure (*c*) as v_1, v_2, v_3 and v_4. The adjacency matrix representing the graph is given by

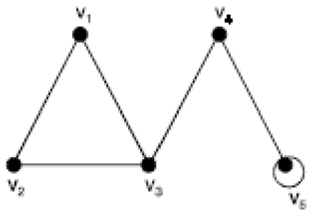
$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Problem 15. *Draw the undirected graph represented by adjacency matrix A given by*

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Solution.

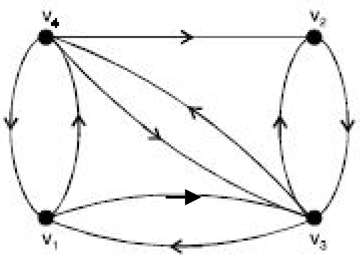
Since the given matrix is a square of order 5, the graph G has five vertices v_1, v_2, v_3, v_4 and v_5.Draw an edge from v_i to v_j where $a_{ij} = 1$.
The required graph is drawn in Figure below.



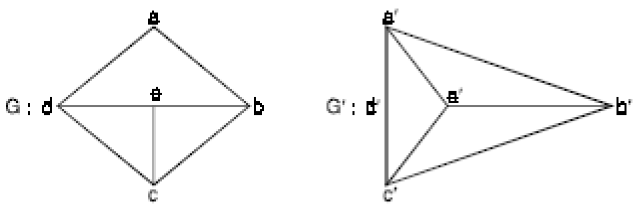
Problem 16. *Draw the digraph G corresponding to adjacency matrix*

$$A = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Solution. Since the given matrix is square matrix of order four, the graph G has 4 vertices v_1, v_2, v_3 and v_4 . Draw an edge from v_i to v_j where $a_{ij} = 1$.
The required graph is shown in Figure below.



Problem 17. *Show that the graphs G and G ' are isomorphic*



Solution. Consider the map f : G →G′ defined as $f(a) = d'$, $f(b) = a'$, $f(c) = b'$, $f(d) = c'$ and $f(e) = e'$

The adjacency matrix of G for the ordering a, b, c, d and e is

$$A(G) = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

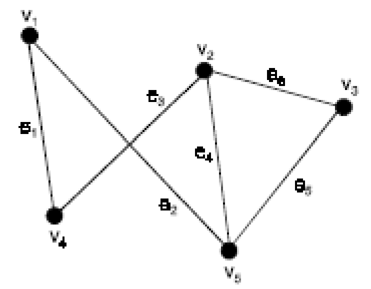
The adjacency matrix of G′ for the ordering d', a', b', c' and e' is

$$A(G') = \begin{array}{c} \\ d' \\ a' \\ b' \\ c' \\ e' \end{array} \begin{array}{c} \begin{array}{ccccc} d' & a' & b' & c' & e' \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

i.e., $A(G) = A(G')$

Therefore G and G' are isomorphic.

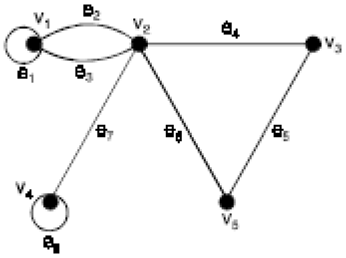
Problem 18. *Represent the graph shown in Figure below, with an incidence matrix.*



Solution. The incidence matrix is

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

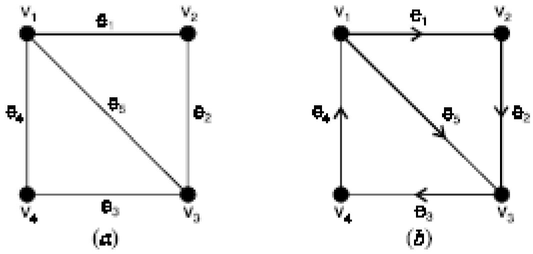
Problem 19. *Represent the Pseudo graph shown in Figure below, using an incidence matrix.*



Solution. The incidence matrix for this graph is

$$
\begin{array}{c}
\\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5
\end{array}
\begin{array}{c}
\begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0
\end{bmatrix}
\end{array}
$$

Problem 20. *Find the incidence matrix to represent the graph shown in Figure below :*



Solution.

The incidence matrix of Figure (*a*) is obtained by entering for row *v* and column *e* is 1 if *e* is incident on *v* and 0 otherwise. The incidence matrix is

$$
\begin{array}{c}
\\ v_1 \\ v_2 \\ v_3 \\ v_4
\end{array}
\begin{array}{c}
\begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\
\begin{bmatrix}
1 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0
\end{bmatrix}
\end{array}
$$

The incidence matrix of the graph of Figure (*b*) is

$$
\begin{bmatrix}
1 & 0 & 0 & -1 & 1 \\
-1 & 1 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & -1 \\
0 & 0 & -1 & 1 & 0
\end{bmatrix}
$$

Problems for practice

1. *Draw the undirected graph G corresponding to adjacency matrix*

WALKS, PATHS AND CYCLES

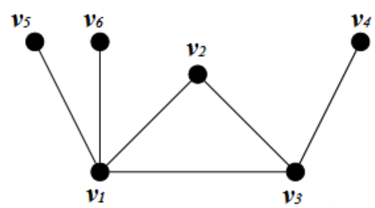
Definition

A walk in G is a sequence of vertices $v_0,\ v_1,\ldots,v_k$ and a sequence of edges$(v_i,\ v_{1+1}) \in E\ (G)$. A walk is a path if all v_i are distinct. v_0 is the initial vertex and v_k is the terminal vertex. A zero length walk is just a single vertex v_0. If for such a path with $k\ \geq\ 2$, $(v_0,\ v_k)$is also an edge in G, then $v_0,\ v_1,\ldots,\ v_k,\ v_0$ is a cycle. For multigraphs, we also consider loops and pairs of multiple edges to be cycles.

Definition

The length of a path, cycle or walk is the number of edges in it.

Example



$v_5v_1v_3v_4$ **is a path of length 3**

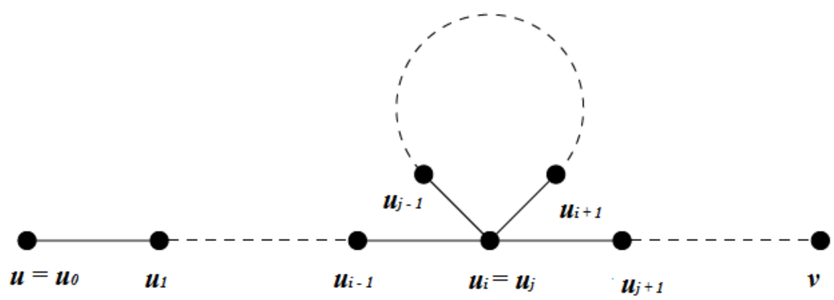
$v_1v_2v_3$ **is a cycle of length 3**

$v_5v_1v_2v_3v_1v_6$ **is a walk of length 5**

Proposition: Every walk from *u* to *v* in G contains a path between *u* and *v*.

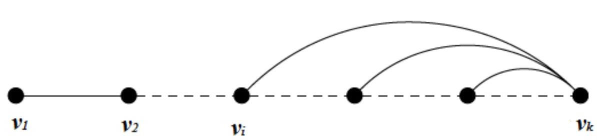
Proof.

By induction on the length *l* of the walk $u = u_0,\ u_1,\ldots,v_l = v$.If $l\ =\ 1$ then our walk is also a path. Otherwise, if our walk is not a path there is $u_i = u_j$with $i\ <\ j$, then $u = u_0,\ u_1,\ u_i,u_{j+1},v$ is also a walk from *u* to *v* which is shorter. We can use induction to conclude the proof.



Proposition: Every G with minimum degree $\delta \geq 2$ contains a path of length δ and a cycle of length at least $\delta + 1$.

Proof. Let $v_1,\ v_2,\ldots,v_k$ be a longest path in G. Then all neighbors of v_k belong to $v_1,v_2,\ldots,v_{k-1}$so $k - 1 \geq \delta$ and $k \geq \delta + 1$, and our path has at least δ edges. Let $i\ (1 \leq i \leq k)$be the minimum index such that $(v_i,\ \ v_k) \in E\ (G)$. Then the neighbors of v_k are among$v_1,\ v_2,\ldots,v_{k-1}$, so $k -\ i \geq \delta$. Then $v_i,\ v_{i+1},\ldots,v_k$ is a cycle of length at least $\delta + 1$.

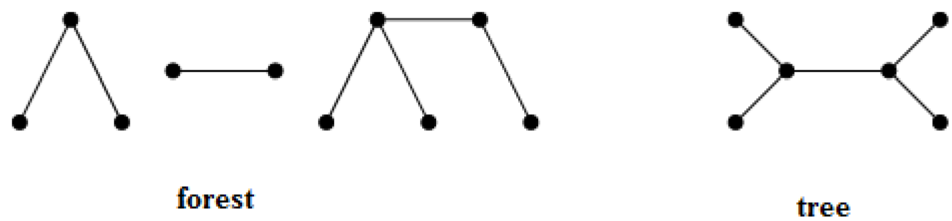


TREES

Definition:

A graph having no cycle is acyclic. A tree is a connected acyclic graph. A leaf (or pendant vertex) is a vertex of degree 1. A forest is an acyclic graph. A tree is a connected forest. A subforest is a subgraph of a forest. A connected subgraph of a tree is a subtree. A spanning tree of a connected graph is a subtree that includes all the vertices of that graph. The edges of a spanning tree are called branches.

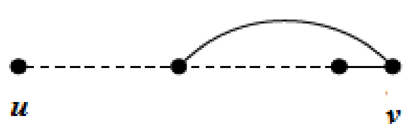
Example:



Lemma: Every finite tree with at least two vertices has at least two leaves. Deleting a leaf from an n-vertex tree produces a tree with $n-1$ vertices.

Proof.

Every connected graph with at least two vertices has an edge. In an acyclic graph, the end points of a maximum path have only one neighbor on the path and therefore have degree 1.Hence the endpoints of a maximum path provide the two desired leaves.



Suppose v is a leaf of a tree G, and let $G' = G - v$. If $u, w \in V(G')$, then no u, w-path P in G can pass through the vertex v of degree 1, so P is also present in G'. Hence G' is connected. Since deleting a vertex cannot create a cycle, G' is also acyclic. We conclude that G' is a tree with $n-1$vertices.

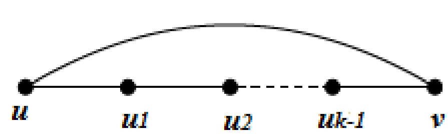
Theorem: For an n-vertex simple graph G (with $n \geq 1$), the following are equivalent (and characterize the trees with n vertices).
(a) G is connected and has no cycles.
(b) G is connected and has $n-1$ edges.
(c) G has $n-1$ edges and no cycles.
(d) For every pair $u, v \in V(G)$, there is exactly one $u, v -$ path in G.
To prove this theorem we will need a small lemma.

Definition: An edge of a graph is a cut-edge if its deletion disconnects the graph.

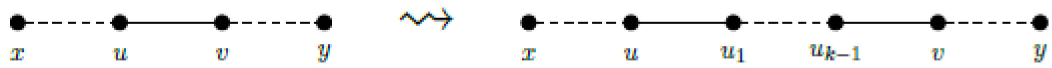
Lemma: An edge contained in a cycle is not a cut-edge.

Proof of the lemma:

Let (u, v) belong to a cycle.



Then any path x ... y in G which uses the edge (u, v) can be extended to a walk in $G-(u, v)$ asfollows:



Proof of Theorem:

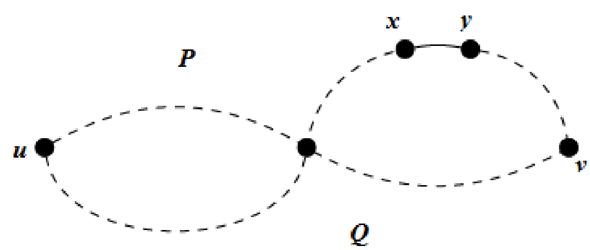
We first demonstrate the equivalence of (a), (b), (c) by proving that any two of {connected, acyclic, $n-1$ edges} implies the third.

(a) ⇒ (b), (c): We use induction on n. For n = 1, an acyclic 1-vertex graph has no edge. For the induction step, suppose n > 1, and suppose the implication holds for graphs with fewer than n vertices. Given G, the Lemma provides a leaf v and states that $G' = G - v$ is acyclic and connected. Applying the induction hypothesis to G'yields $e(G') = n-2$, and hence$e(G) = n-1$.

(b) ⇒ (a), (c): Delete edges from cycles of G one by one until the resulting graph G' is acyclic. By Lemma, G is connected. By the paragraph above, G' has $n-1$ edges. Since this equals$|E(G)|$, no edges were deleted, and G itself is acyclic.

(c) ⇒ (a), (b): Suppose G has k components with orders $n_1, . . . n_k$. Since G has no cycles, each component satisfies property (a), and by the first paragraph the *i*th component has $n_i - 1$ edges. Summing this over all components yields $e(G) = \sum(n_i - 1) = n - k$. We are given $e(G) = n - 1$, so $k = 1$, and G is connected.

(a) ⇒ (d): Since G is connected, G has at least one u, v −path for each pair $u, v \in V(G)$. SupposeG has distinct u, v −paths P and Q. Let $e = (x, y)$ be an edge in P but not in Q. The concatenation of P with the reverse of Q is a closed walk in which e appears exactly once. Hence, $(P \cup Q) - e$ is an x, y −walk not containing e. Thus we have a cycle with e and contradicts the hypothesis that G is acyclic. Hence G has exactly one u, v −path.



(d) ⇒ (a): If there is a u; v-path for every u; v ε V (G), then G is connected. If G has a cycle C, then G has two paths between any pair of vertices on C.

Definition:

Given a connected graph G, a spanning tree T is a subgraph of G which is a tree and contains every vertex of G.

Corollary:

(a) Every connected graph on n vertices has at least $n-1$ edges and contains a spanning tree;

(b) Every edge of a tree is a cut-edge;

(c) Adding an edge to a tree creates exactly one cycle.

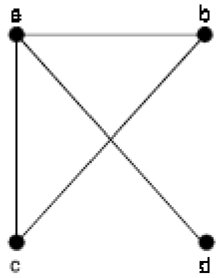
Proof.

(a) Delete edges from cycles of G one by one until the resulting graph G_0 is acyclic. By Lemma, G is connected. The resulting graph is acylic so it is a tree. Therefore G had at least $n-1$edges and contains a spanning tree.

(b) Note that deleting an edge from a tree T on n vertices leaves $n-2$ edges, so the graph is disconnected by (a).

$$A = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 2 & 2 \\ 0 & 1 & 2 & 0 \end{bmatrix}$$

2. *Use an adjacency matrix to represent the graph shown in Figure below*

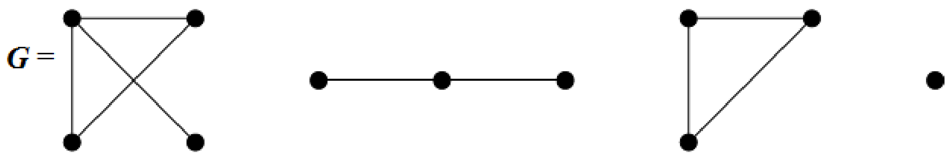


3. *Draw a graph with the adjacency matrix*

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Definition:

A (connected) component of G is a connected subgraph that is maximal by inclusion. We say G is connected if and only if it has one connected component. The graph G which is given below has 4 connected components.



Proposition: A graph with n vertices and m edges has at least $n - m$ connected components.

Proof.

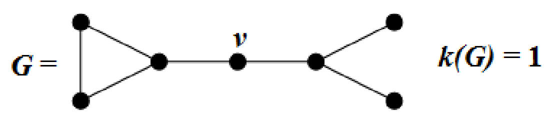
Start with the empty graph (which has n components), and add edges one-by-one. Note that adding an edge can decrease the number of components by at most 1.

Definition: (Vertex connectivity)

A vertex cut in a connected graph $G = (V, E)$ is a set $S \subseteq V$ such that $G \setminus S = G[V \setminus S]$ has more than one connected component. A cut vertex is a vertex v such that $\{v\}$is a cut.

Definition:

G is called k-connected if $|V(G)| > k$ and if $G \setminus X$ is connected for every set $X \subseteq V$with $|X| < k|$. In other words, no two vertices of G are separated by fewer than k other vertices. Every (non-empty) graph is 0-connected and the 1-connected graphs are precisely the non-trivial connected graphs. The greatest integer k such that G is $k-$connected is the connectivity $k(G)$of G. For example, if $G = K_n$, then $k(G) = n - 1$. In the below example, deleting v disconnects G, so v is a cut vertex.



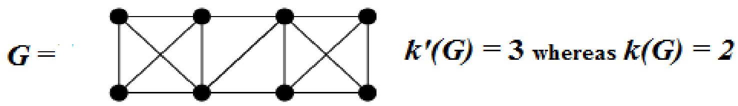
Proposition: For every graph G, $k(G) \leq \delta(G)$.

Proof.

Let $v \in V(G)$ be a vertex of minimum degree $d(v) = \delta(G)$. Then deleting $N(v)$ disconnects v from the rest of G.

Definition: (Edge connectivity)

A disconnecting set of edges is a set $S \subseteq E(G)$ such that $G \setminus F$has more than one component. Given $S, T \subseteq V(G)$ the notation [S, T] specifies the set of edges having one end point in S and the other in T. An edge cut is an edge set of the form [S, $\bar{S}$], where S is a non-empty proper subset of V (G). A graph is k-edge-connected if every disconnecting set has at least k edges. The edge-connectivity of G, written $k'(G)$, is the minimum size of a disconnecting set. One edge disconnecting G is called a bridge. For example, if $G = K_n$, then $k'(G) = n - 1$.



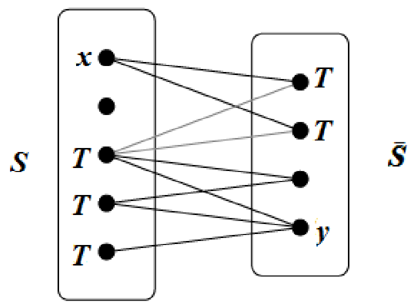
Remark: An edge cut is a disconnecting set but not the other way around. However, everyminimal disconnecting set is a cut.

Theorem:$k(G) \leq k'(G) \leq \delta(G)$.

Proof.

The edges incident to a vertex v of minimum degree, form a disconnecting set, hence $k'(G) \leq \delta(G)$. It remains to show that $k(G) \leq k'(G)$. Suppose$|G| > 1$ and [S, S] is a minimum edge cut, having size $k'(G)$.

If every vertex of S is adjacent to every vertex of $\bar{S}$ and $|G| = |V(G)| = n$, then $k'(G) = |S||\bar{S}| = |S|(|G| - |S|)$. This expression is minimized at $|S| = 1$. By definition,$k(G) \leq |G| - 1$, so the inequality holds.



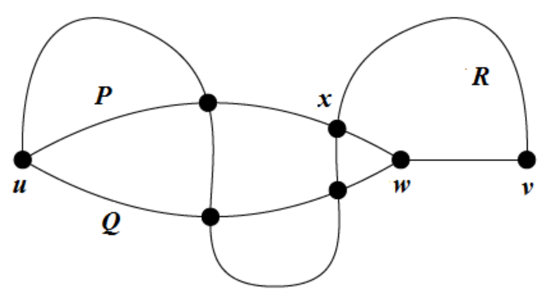
Hence we may assume there exists $x \in S$, $y \in \bar{S}$ with x not adjacent to y. Let T be the vertex set consisting of all neighbors of x in S and all vertices of $S \backslash x$ that have neighbours in S (illustrated below). Deleting T destroys all the edges in the cut [S, S] (but does not delete x or y), so T is a separating set. Now, by the definition of T we can injectively associate at least one edge of $[S,\bar{S}\,]$ to each vertex in T, so .$k(G) \leq |T| \leq |[S,\bar{S}]| = k'(G)$.

Definition: Two paths are internally disjoint if neither contains a non-endpoint vertex of the other. We denote the length of the shortest path from u to v (the distance from u to v) by d(u, v).

Theorem: (Whitney 1932).A graph G having at least three vertices is 2-connected if and only if each pair $u, v \in V(G)$ is connected by a pair of internally disjoint $u, v -$ paths in G.

Proof.

When G has internally disjoint u, v −paths, deletion of one vertex cannot separate u from v. Since this is given for every u, v, the condition is sufficient. For the converse, suppose that G is2-connected. We prove by induction on $d(u, v)$ that G has two internally disjoint u, v paths. When $d(u, v) = 1$, the graph $G \setminus (u, v)$ is connected, since $k'(G) \geq k(G) = 2$. A $u, v -$ path in $G \setminus (u, v)$ is internally disjoint in G from the u, v −path consisting of the edge (u, v) itself.



For the induction step, we consider $d(u,v) = k > 1$ and assume that G has internally disjoint x,y −paths whenever $1 \leq d(x,y) \leq k$. Let w be the vertex before v on a shortest u,v −path. We have $d(u,w) = k - 1$, and hence by the induction hypothesis G has internally disjoint u,w − paths P andQ. Since $G \setminus w$ is connected, $G \setminus w$ contains a u,v −path R. If this path avoids P or Q, we are finished, but R may share internal vertices with both P and Q. Let x be the last vertex of R belonging to $P \cup Q$. Without loss of generality, we may assume, $x \in P$. We combine the u,x −subpath of P with the x,v −subpath of R to obtain a u,v −path internally disjoint from $Q \cup \{(w,v)\}$.

Corollary: G is 2-connected and$|V(G)| \geq 3$ if and only if every two vertices in G lie on a common cycle.

EULERIAN AND HAMILTONIAN PATHS

Definition: A trail is a walk with no repeated edges.

Definition: An Eulerian trail in a graph $G = (V,E)$ is a walk in G passing through every edge exactly once. If this walk is closed (starts and ends at the same vertex) it is called an Eulerian tour.

Theorem: A connected graph has an Eulerian tour if and only if each vertex has even degree.
In order to prove this theorem we use the following lemma.

Lemma: Every maximal trail in a graph where all the vertices have even degree is a closed trail.

Proof.
Let T be a maximal trail. If T is not closed, then T has an odd number of edges incident to the final vertex v. However, as v has even degree, there is an edge incident to v that is not in T. This edge can be used to extend T to a longer trail, contradicting the maximality of T.

Proof of Theorem
To see that the condition is necessary, suppose G has an Eulerian tour C. If a vertex v was visited k times in the tour C, then each visit used 2 edges incident to v (one in comingedge and one outgoing edge). Thus, $d(v) = 2k$, which is even.

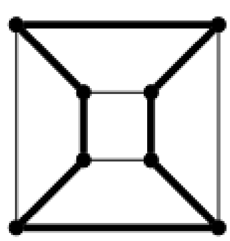
To see that the condition is sufficient, let G be a connected graph with even degrees. Let $T = e_1 e_2 \ldots e_l$ (where $e_i = (v_{i-1}, v_i)$)be a longest trail in G. Then, by Lemma, T is closed, that is, $v_0 = v_l$. If T does not include all the edges of G then, since G is connected, there is an edge outside of T such that $e = (u, v_i)$ for some vertex v_i in T. But then $T' = ee_{i+1} \ldots e_l e_1 e_2 \ldots e_i$ is a trail in G which is longer than T, contradicting the fact that T is a longest trail in G. Thus, we conclude that T includes all the edges of G and so it is an Eulerian tour.

HAMILTON PATHS AND CYCLES

Definition: A Hamilton path/cycle in a graph G is a path/cycle visiting every vertex of G exactly once. A graph G is called Hamiltonian if it contains a Hamilton cycle.

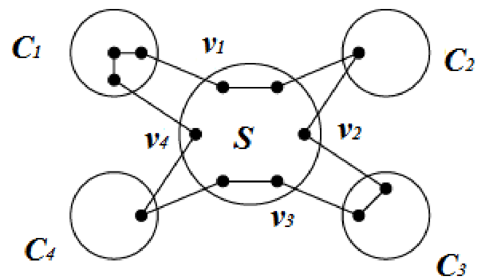
Hamilton cycles were introduced by Kirkman in 1985, and were named after Sir William Hamilton, who produced a puzzle whose goal was to find a Hamilton cycle in a specific graph.

Example: Hamilton cycle in the skeleton of the 3-dimensional cube.



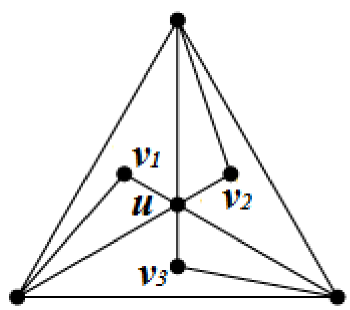
Proposition 5.3.
Theorem: If G is Hamiltonian then for any set $S \subseteq V(G)$ the graph $G \setminus S$has at most $|S|$connected components.
Proof.

Let $C_1, C_2, \dots, C_k$ be the components of $G \setminus S$. Imagine that we are moving along a Hamilton cycle in some order, vertex-by-vertex (in the picture below, we are moving clockwise, starting from some vertex in C_1, say). We must visit each component of $G \setminus S$at least once, when we leaveC_i forthe first time, let v_i be the subsequent vertex visited (which must be in S). Each v_i must be distinct because a cycle cannot intersect itself. Hence, S must have at least as many vertices as the number of connected components of $G \setminus S$.



Example:

The condition in Proposition is not sufficient to ensure that a graph is Hamiltonian. The graph G above satisfies the condition of Proposition, but is not Hamiltonian. Indeed, one would need to include all the edges incident to the vertices v_1, v_2 and v_3 in a Hamiltoncycle of G, however, in that case the vertex u would have degree at least 3 in that Hamilton cycle, which is impossible. We also give some sufficient conditions for Hamiltonicity.



Theorem: (Dirac 1952). If G is a simple graph with $n \geq 3$ vertices and if $\delta(G) \geq \frac{n}{2}$, then G is Hamiltonian.
Proof.

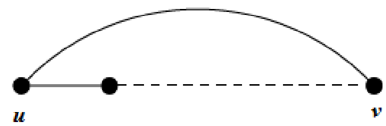
The condition that $n \geq 3$ must be included since K_2 is not Hamiltonian but satisfies $\delta(G) = \frac{|K_2|}{2}$. If there is a non-Hamiltonian graph satisfying the hypotheses, then adding edges cannot reduce the minimum degree, so we may restrict our attention to maximal non-Hamiltonian graphs G with minimum degree at least $\frac{n}{2}$. By “maximal" we mean that for every pair (u, v) of non-adjacent vertices of G, the graph obtained from G by adding the edge $e = (u, v)$ is Hamiltonian.

The maximality of G implies that G has a Hamilton path, say from $u = v_1$, to $v = v_n$, because every Hamilton cycle in $G \cup \{e\}$ must contain the new edge e. We use most of this path $v_1, v_2, \dots, v_n$with a small switch, to obtain a Hamilton cycle in G. If some neighbor of u immediately follows a neighbor of v on the path, say (u; vi+1) ∈ E(G) and (v; vi) ∈ E(G), then G has the Hamilton cycle$(u, v_{i+1}, v_{i+2}, \dots, v_{n-1}, v, v_i, v_{i-1}, \dots, v_2)$ shown below.

To prove that such a cycle exists, we show that there is a common index in the sets S and T defined by $S = \{i: (u, v_{i+1} \in E(G)\}$ and T= $\{i: (v, v_i \in E(G)\}$. Summing the sizes of these sets, yields$|S \cup T| + |S \cap T| = |S| + |T| = d(u) + d(v) \geq n$. Neither S nor T contains the index n. This implies that $|S \cup T| < n$, and hence $|S \cap T| \geq 1$, as required. This is a contradiction.

It can be observed that this argument uses only that $d(u) + d(v) \geq n$. Therefore, we can weaken the requirement of minimum degree $\frac{n}{2}$ to require only that $d(u) + d(v) \geq n$ whenever *u* is not adjacent to *v*.

(c) Let $u, v \in T$. There is a unique path in T between u and v, so adding an edge (u, v) closes this path to a unique cycle.



Theorem: A connected graph has at least one spanning tree.

Proof.

Consider the connected graph G with n vertices and m edges. If $m = n - 1$, then G is a tree. Since G is connected, $m \geq n - 1$. We still have to consider the case $m \geq n$, where there is a circuit in G. We remove an edge e from that circuit. G − e is now connected. We repeat until there are $n - 1$ edges. Then, we are left with a tree.

Theorem: If a tree is not trivial, then there are at least two pendant vertices.

Proof.

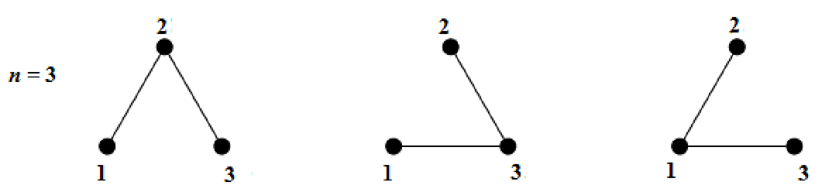
If a tree has n ≥2 vertices, then the sum of the degrees is 2(n − 1). If every vertex has a degree ≥ 2, then the sum will be $\geq 2n$. On the other hand, if all but one vertex have degree ≥ 2, then the sum would be $\geq 1 + 2(n - 1) = 2n - 1$. This is because a cut vertex of a tree is not a pendant vertex. A forest with k components is sometimes called a k-tree. (So a 1-tree is a tree.)

Theorem (Cayley's Formula). There are n^{n-2}trees with vertex set n.

Question: What is the number of spanning trees in a labeled complete graph on n vertices?

By Cayley's formula, it is n^{n-2}.

Example:



Theorem: If G is a tree, then the number of edges in G= $n - 1$.

Proof.

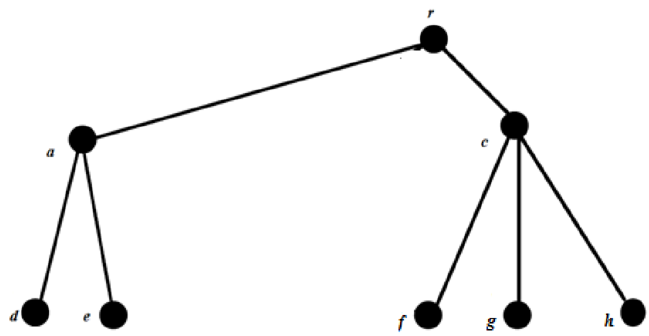
Let us denote the number of edges in G by m. By induction on n, when n = 1, G is isomorphic to K_1 and so the number of edges in G is $m = 0 = n - 1$. Suppose the theorem is true for all trees on fewer than v vertices and let G be a tree on $n \geq 2$ vertices. Let $(u, v) \in E(G)$, then $G - (u, v)$ contains no $u, v -$ path, since (u, v) is the unique $u, v -$ path in G. Thus $G - (u, v)$ is disconnected so $\omega(G - uv) = 2$.The components G_1 and G_2 of $G - (u, v)$, being acyclic are trees. Moreover, each has fewer than n vertices. Therefore by induction hypothesis, $E(G_i) = V(G_i) - 1$, for $i = 1, 2$. Thus$E(G) = E(G_i) + E(G_i) + 1 = V(G_1) + V(G_2) + 1 = V(G) - 1 = n - 1$.

CONNECTIVITY

Definition:

A graph G is connected if, for all pairs $u, v \in V(G')$, there is a path in G from u to v.

Note that it suffices for there to be a walk from u to v, by Proposition



Definition: The left (right) subtree of a vertex v in a binary tree is the binary subtree spanning the left (right)-child of v and all of its descendants.

Theorem: The complete binary tree of height h has $2^{h+1} - 1$ vertices.

Corollary: Every binary tree of height h has at most $2^{h+1} - 1$vertices.

Expression Trees

An expression tree is a special type of a binary tree that represents an algebraic expression in such a way that stores its structure and shows how the order of operations applies. This is a very important type of a tree in computer science. We're interested in a few different operators. We break these operators down into two categories:

- Binary Operators - operators that take two inputs
  - +
  - - (here, subtraction)
  - *
  - / (both integer and floating-point division)
  - % (modulus)
  - ^ or ** (exponentiation)

- Unary Operators - operators that take one input
  - - (here, negation)

Note that we don't mention parentheses. The expression tree's structure removes the need to talk about parentheses, as the structure encodes precedence.

When we have a single expression based on a binary operator, we draw the expression tree as follows:

- The operator is the root of the tree.
- The operands are the children. Because some operations are *not* commutative, order does matter. The operand before the operator is the left child and the operand after the operator is the right child. Thus, we get a tree with a root and two children. For example see figure (a).

When we have a single expression based on a unary operator, we draw the expression tree as follows:

- The operator is the root of the tree.
- The operand is the child.

Thus, we get a tree with a root and one child. (It's really more of a linear structure than a tree, but it does fit the definition of a tree. We'll find that these kinds of trees are interesting when we join them together as part of more complicated expressions.) The Expression tree for –a is in figure (e). Note that we could treat negation as multiplication by -1 and eliminate the need for unary trees if we'd like to have all nodes in our tree having exactly 2 children (or no child). When we wish to work with more complicated expressions, we invoke the recursive nature of binary trees. When an operand is an expression rather than a single variable or constant, we simply put the expression tree for that expression in lieu of theoperand. Figures (b), (c) and (d) are examples of such expression trees.

(a) a / b **(b)** $a * b + c$ **(c)** $a * (b + c)$ **(d)** $a * (b + c * d) / e$ **(e)** $- a$