CS6109 – GRAPH THEORY

Module – 1

Presented By
Dr. S. Muthurajkumar,
Assistant Professor,
Dept. of CT, MIT Campus,
Anna University, Chennai.

INTRODUCTION

- Introduction
- Graph Terminologies
- Types of Graphs
- Isomorphism
- Isomorphic Graphs
- Operations on graphs
- Degree sequences
- Euler graph
- Hamiltonian Graph
- Related theorems

GRAPH Definition

- A graph is simply a collection of nodes plus edges
- Linked lists, trees, and heaps are all special cases of graphs
- The nodes are known as vertices (node = “vertex”)
- **Formal Definition: A graph *G* is a pair (*V*, *E*)**

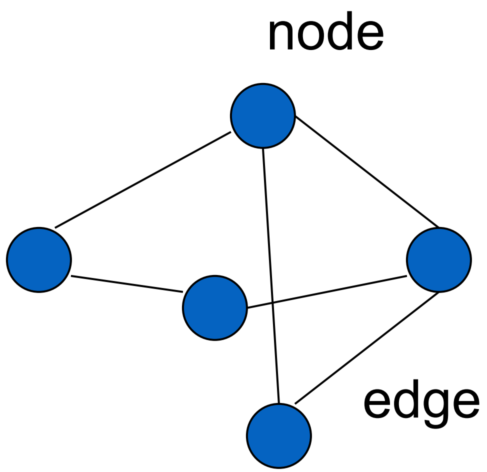
G = (V, E)

where

- V is a set of vertices or nodes
- E is a set of edges that connect vertices
- Graph size parameters: $\boldsymbol{n} = |V|$, $\boldsymbol{m} = |E|$.

Vertex and Edge

- Vertex/Node
  - Basic Element
  - Drawn as a node or a dot
  - Vertex set of G is usually denoted by V(G), or V or VG.
- Edge/Arcs
  - A set of two elements
  - Drawn as a line connecting two vertices, called end vertices, or endpoints.
  - The edge set of G is usually denoted by E(G), or E or EG.
- Neighborhood
  - For any node v, the set of nodes it is connected to via an edge is called its neighborhood and is represented as N(v)



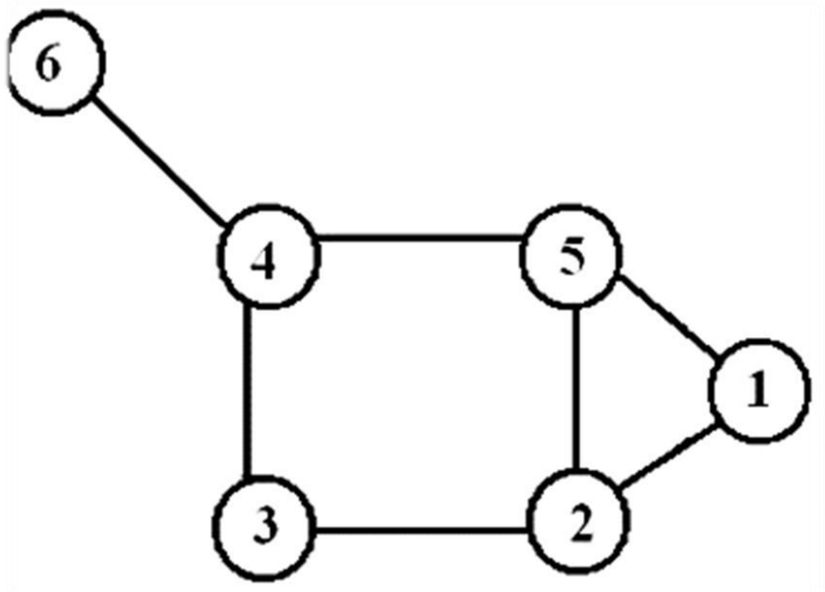
Graph Example

➔n: = 6, m:= 7

➔Vertices (V): ={1,2,3,4,5,6}

➔Edge (E): = {{1,2},{1,5},{2,3},{2,5}{3,4},{4,5},{4,6}}

➔N(4):= Neighborhood (4) = {6,5,3}



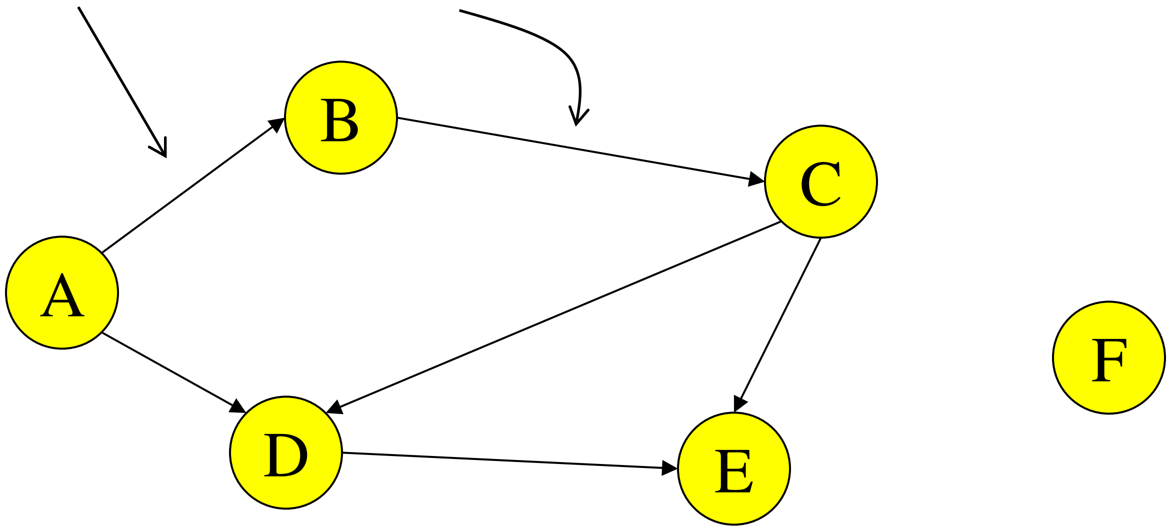
Graph Example

Here is a directed graph G = (V, E)

➔Each edge is a pair (v1, v2), where v1, v2 are vertices in V

➔V = {A, B, C, D, E, F}

➔E = {(A,B), (A,D), (B,C), (C,D), (C,E), (D,E)}



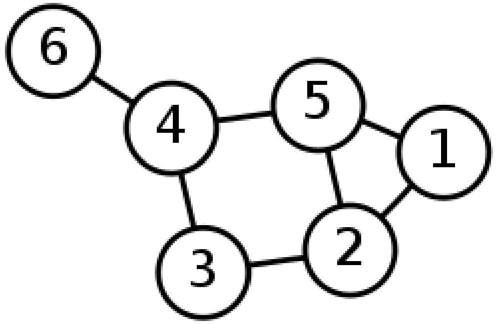
Graph Theory

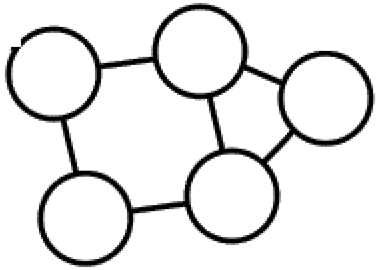
- Graphs are used to model pair wise relations between objects
- Generally a network can be represented by a graph
- Many practical problems can be easily represented in terms of graph theory

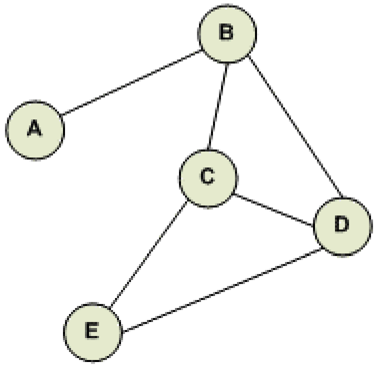
Graph- Varities

- Nodes
  - Labeled or unlabeled
- Edges
  - Directed or undirected
  - Labeled or unlabeled

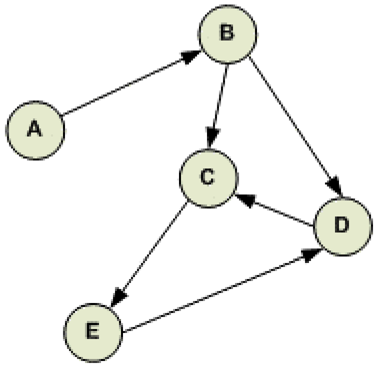
Labeled and unlabeled





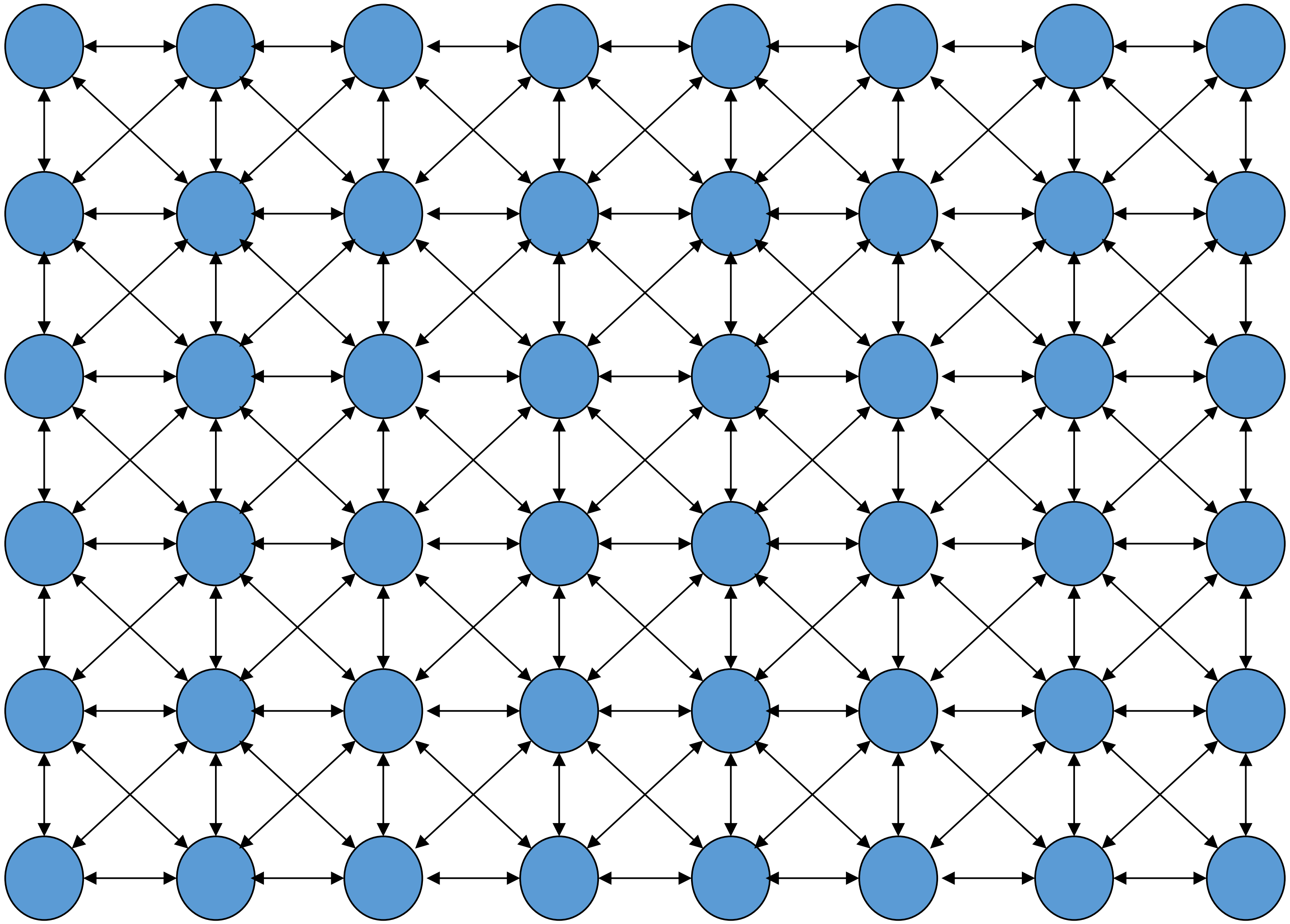


Undirected Graph

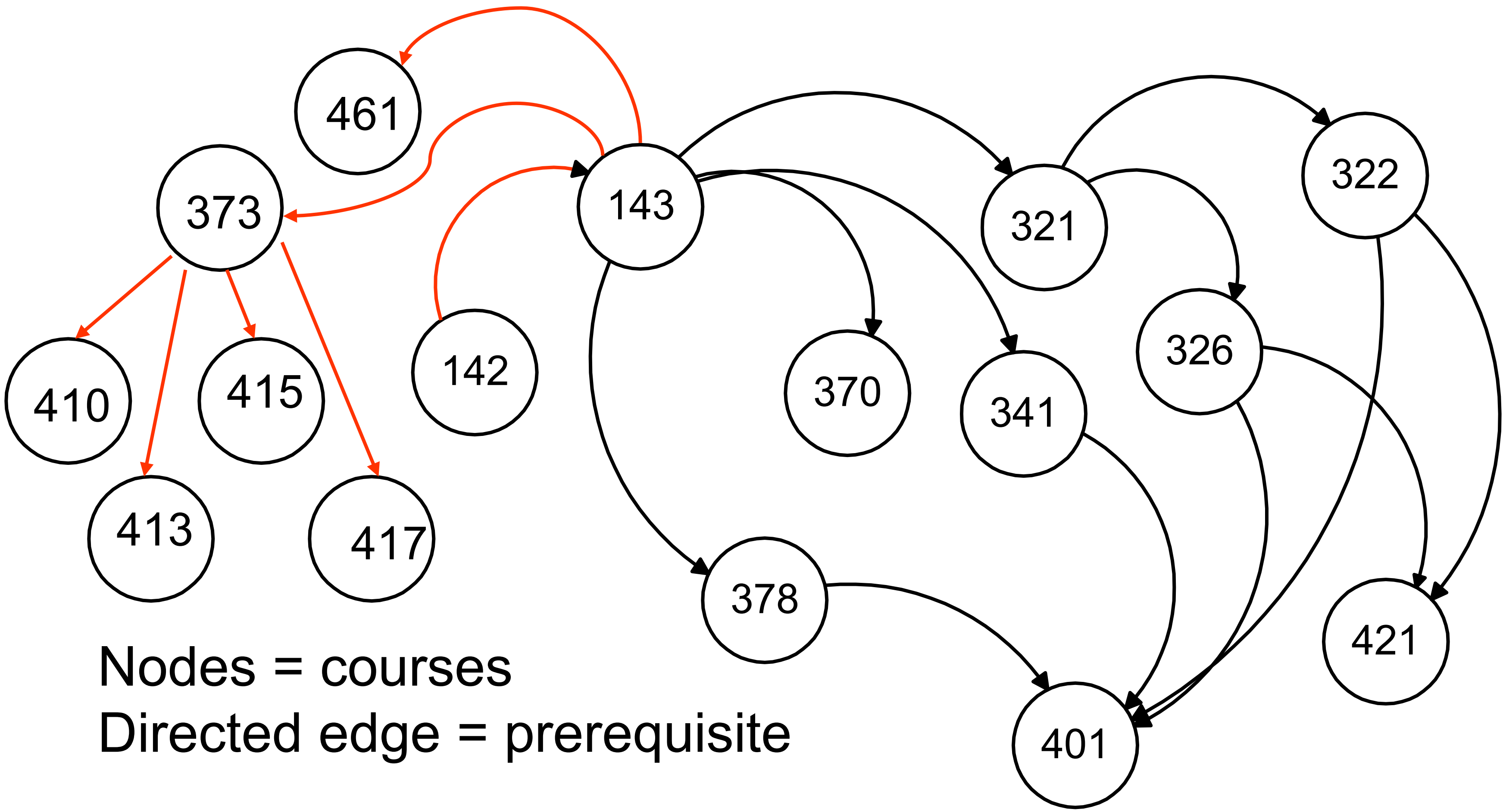


Directed Graph

Regular Graph

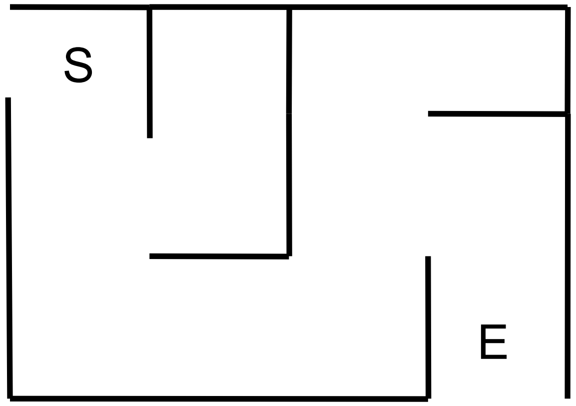


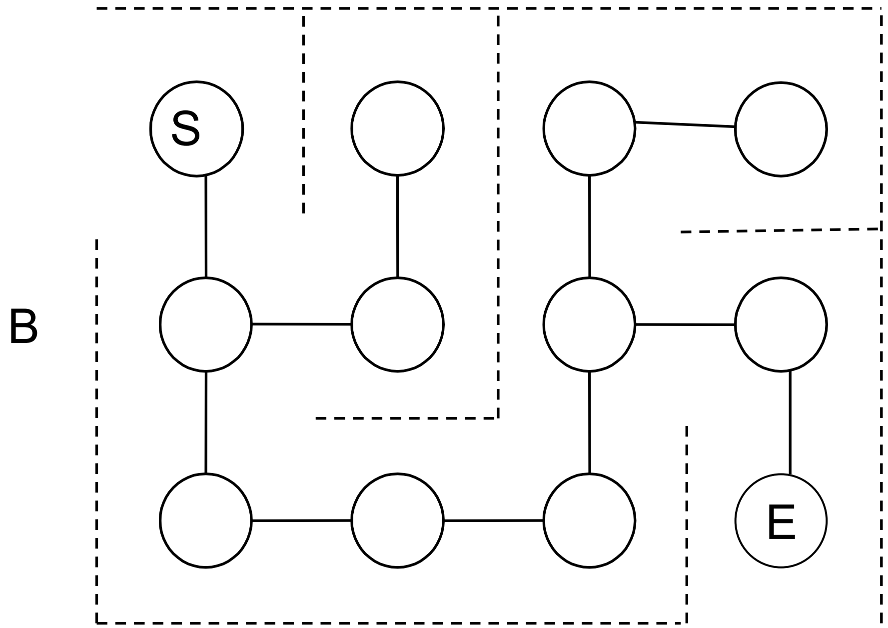
Course Prerequisites



Nodes = courses
Directed edge = prerequisite

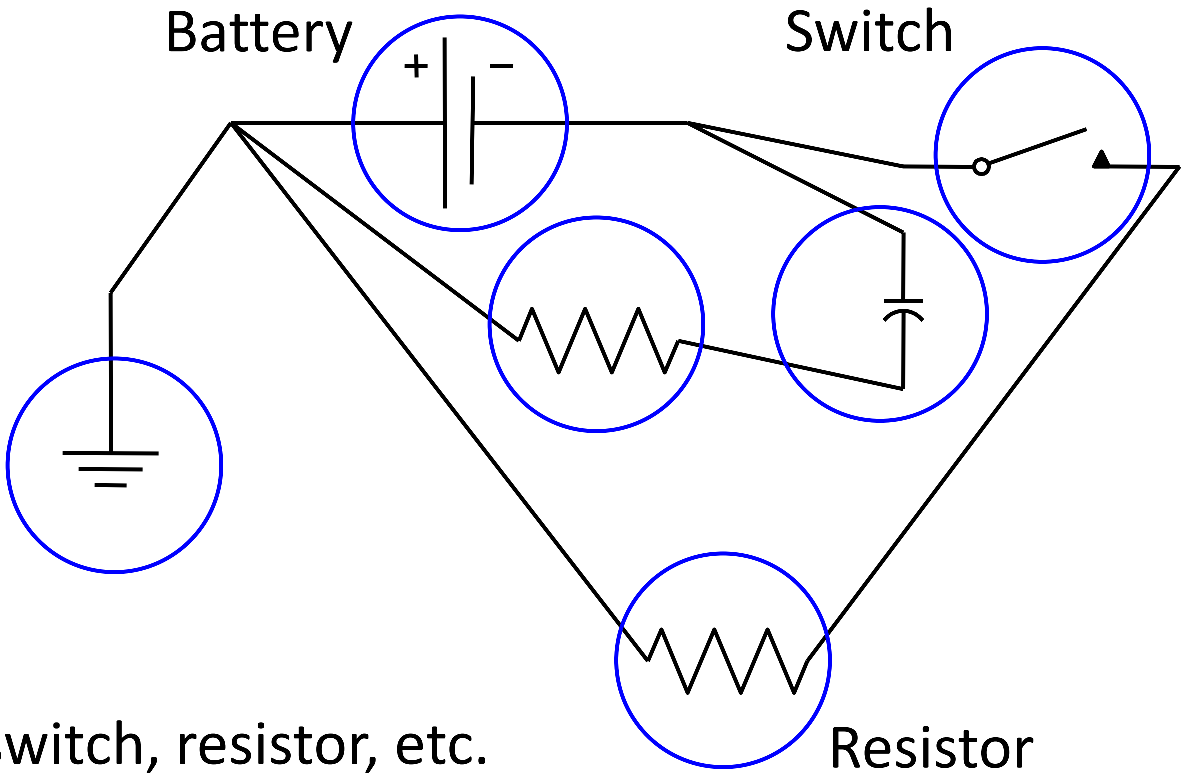
Representing a Maze





Nodes = rooms
Edge = door or passage

Representing Electrical Circuits



Nodes = battery, switch, resistor, etc.
Edges = connections

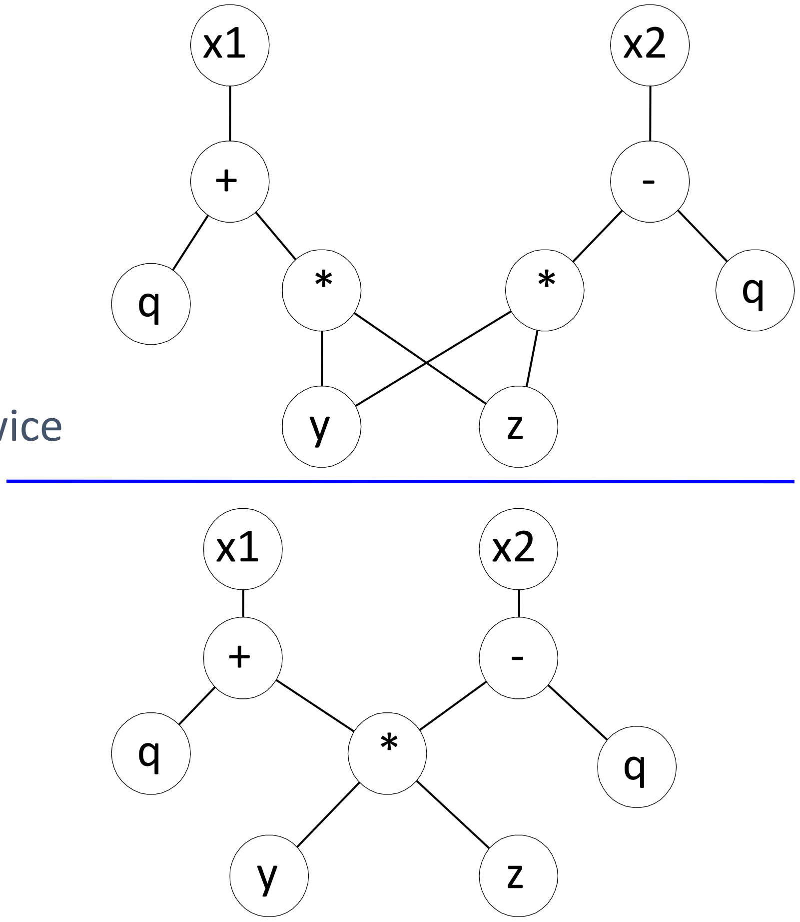
Program statements

x1=q+y*z
x2=y*z-q

Naive:

y*z calculated twice

common
subexpression
eliminated:



Nodes = symbols/operators
Edges = relationships

Precedence

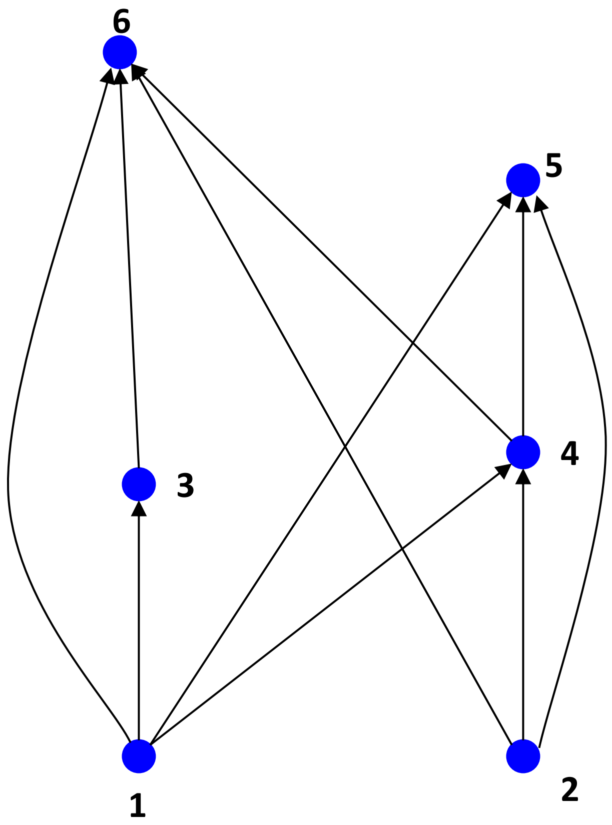
S_1	a=0;
S_2	b=1;
S_3	c=a+1
S_4	d=b+a;
S_5	e=d+1;
S_6	e=c+d;

Which statements must execute before S_6?

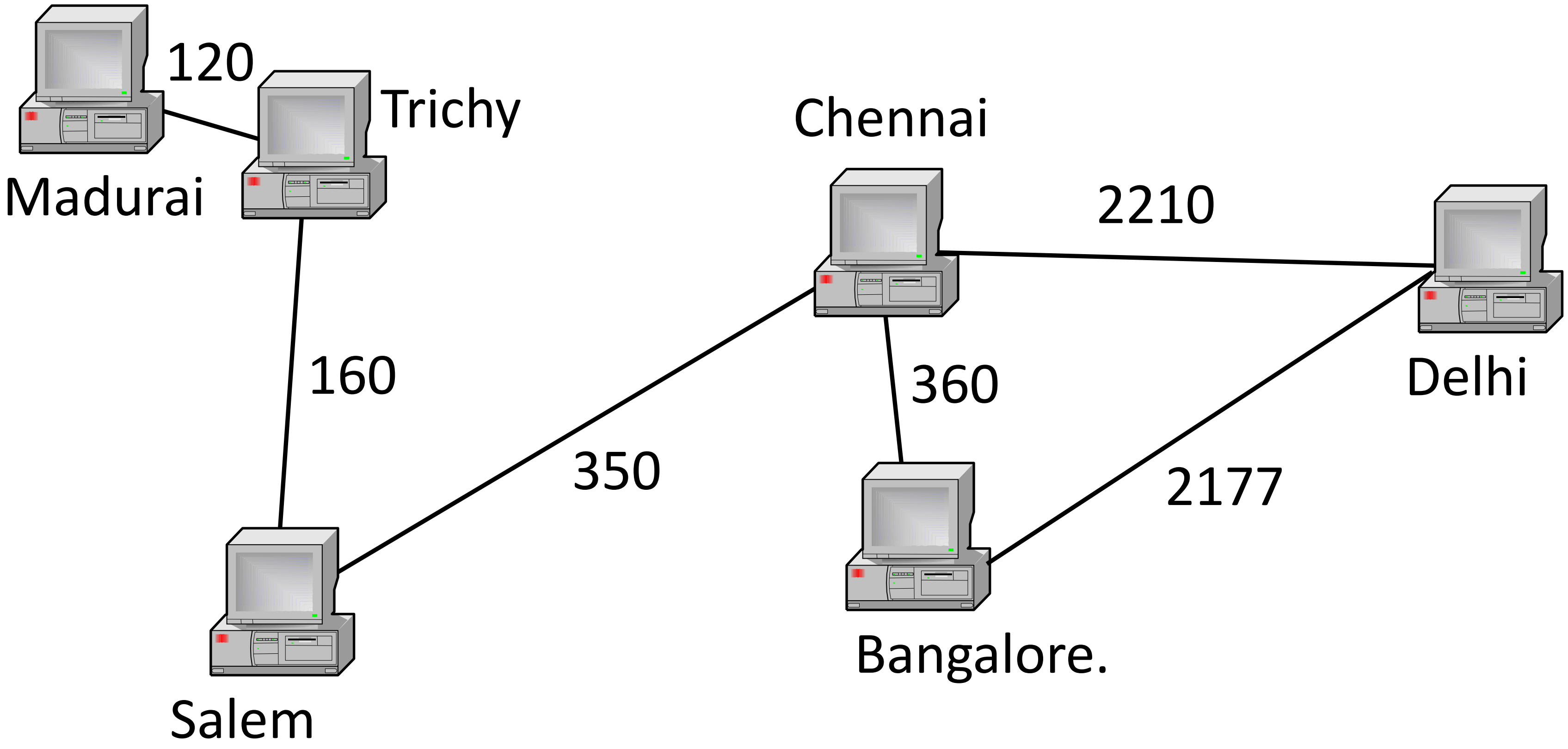
S_1, S_2, S_3, S_4

Nodes = statements

Edges = precedence requirements

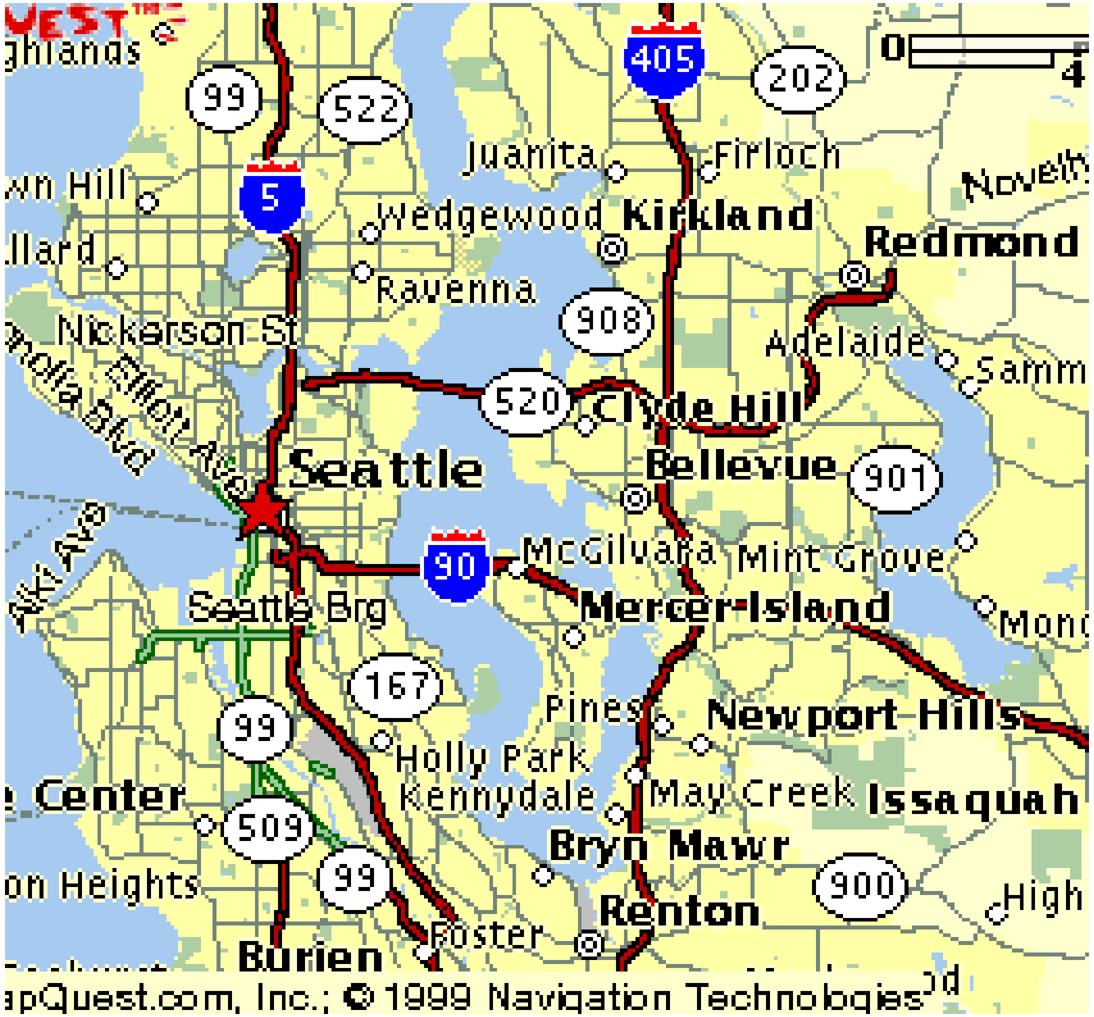


Information Transmission in a Computer Network



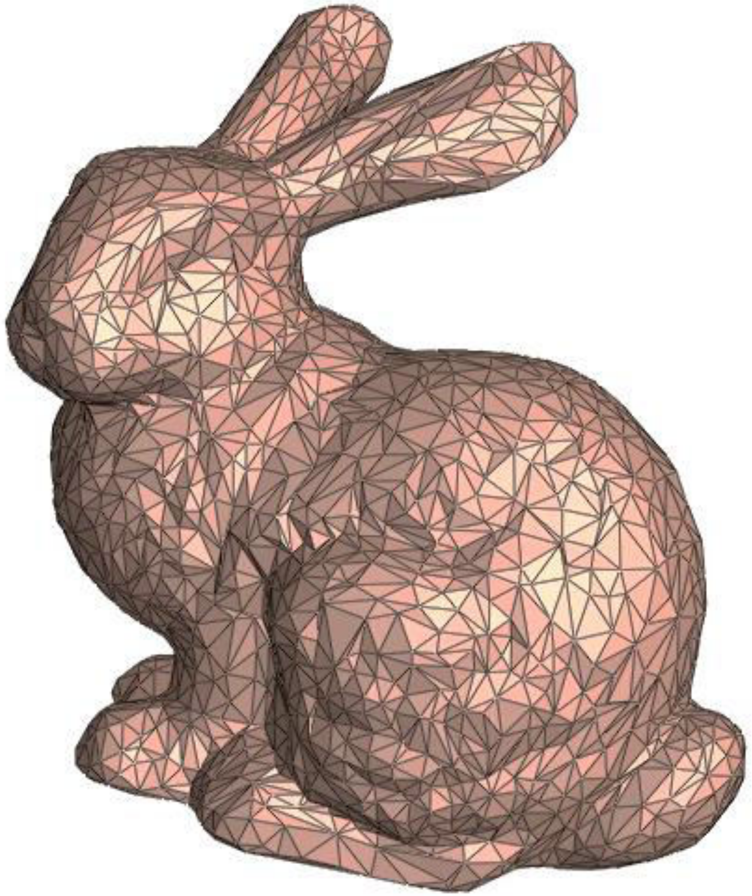
Nodes = computers
Edges = transmission rates

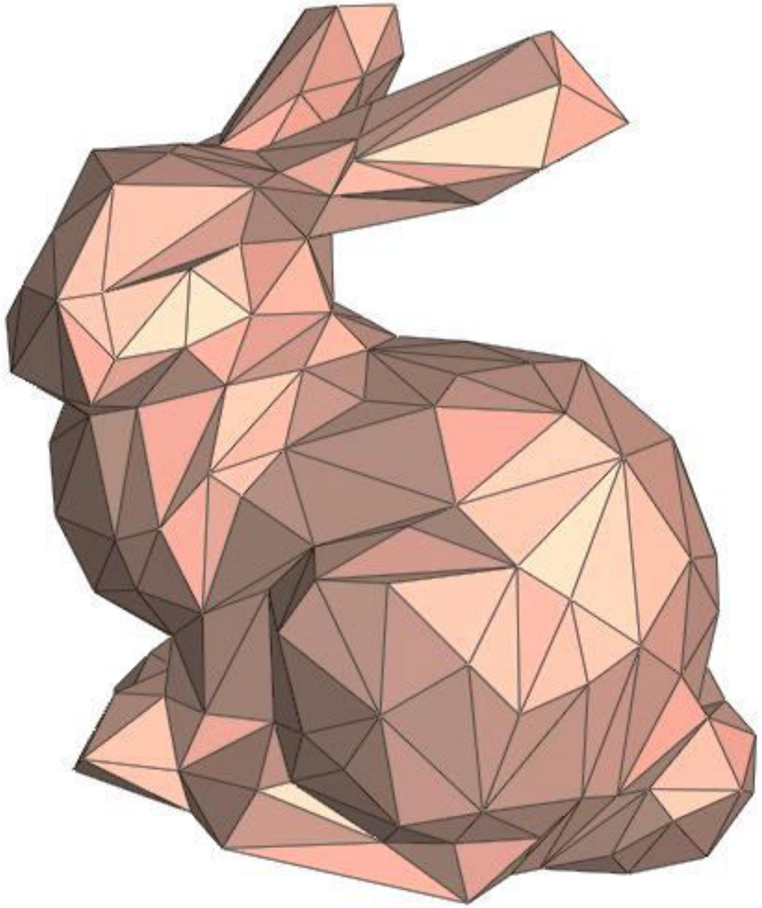
Traffic Flow on Highways



Nodes = cities
Edges = # vehicles on connecting highway

Polygonal Meshes



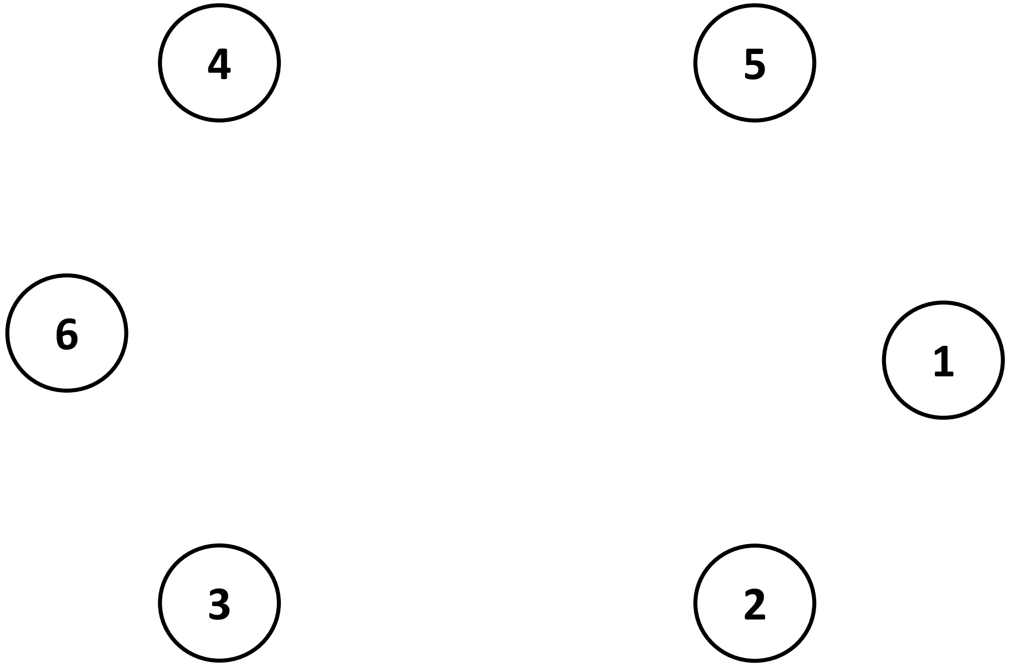


Types of Graph

- Undirected
  - Pairs of nodes
- Directed
  - ordered pairs of nodes.
  - Directed edges have a source (head, origin) and target (tail, destination) vertices
- Weighted
  - usually weight is associated
- Un Weighted

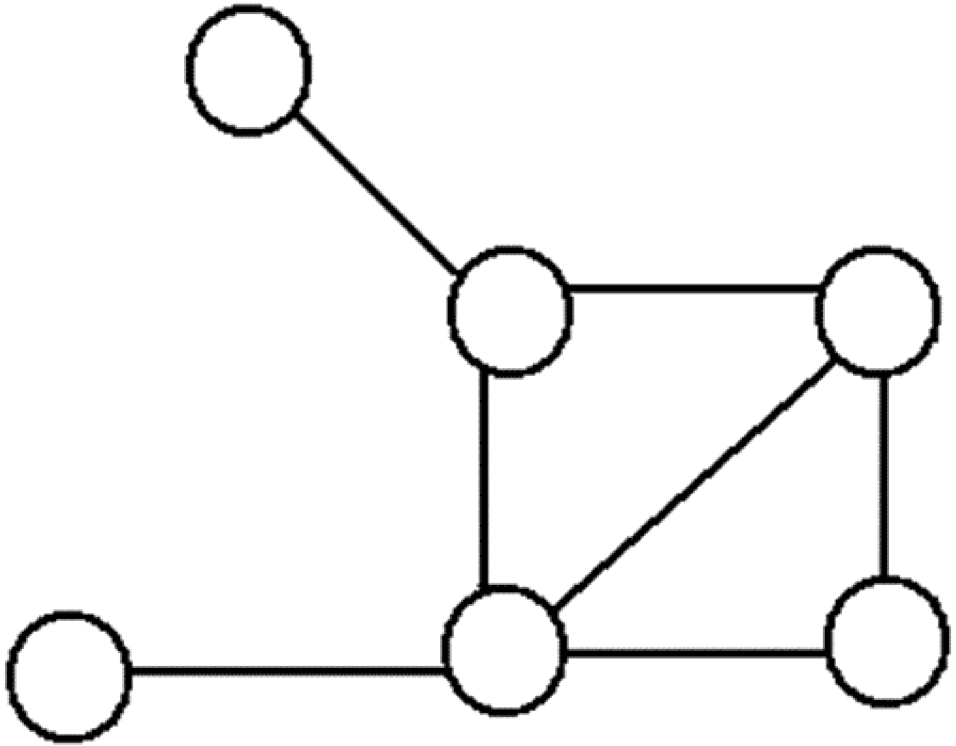
Empty Graph / Edgeless graph

- No Edge
- Null graph
  - No nodes
  - Obviously no edge



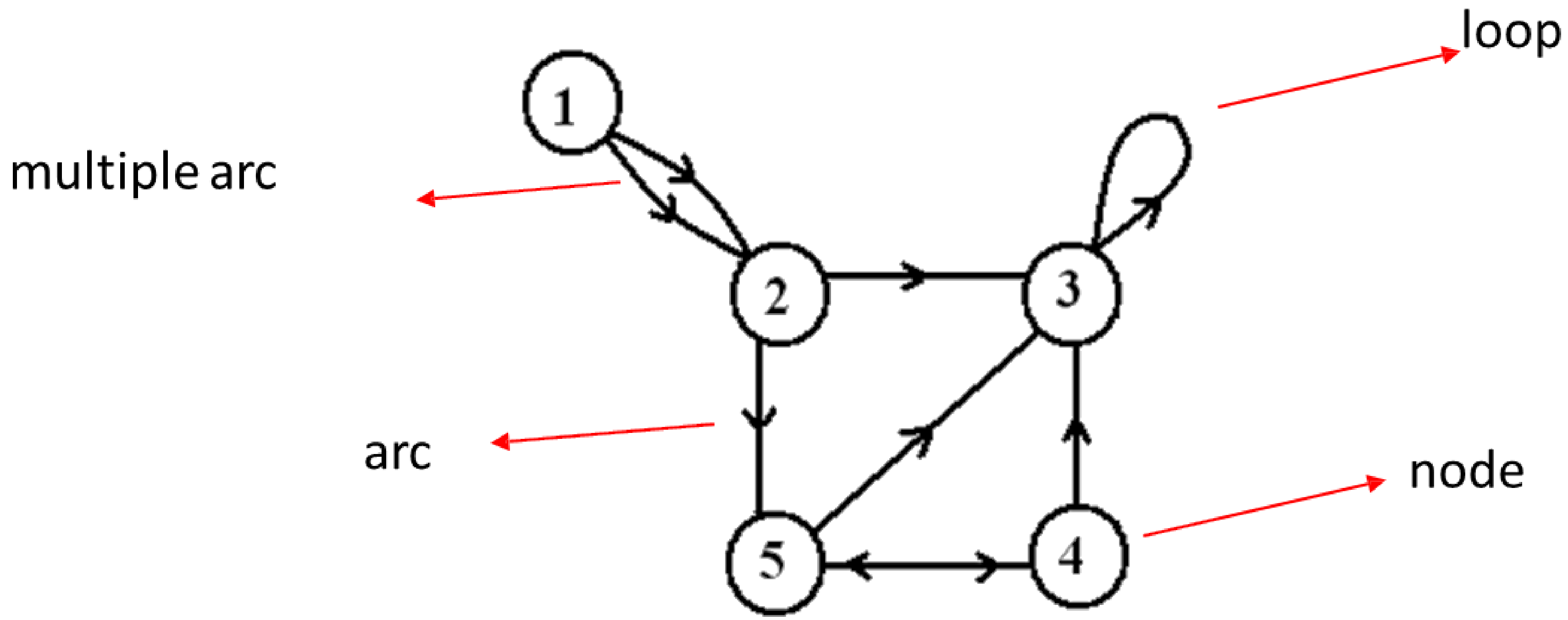
Simple Graph (Undirected)

- Simple Graph are undirected graphs without loop or multiple edges



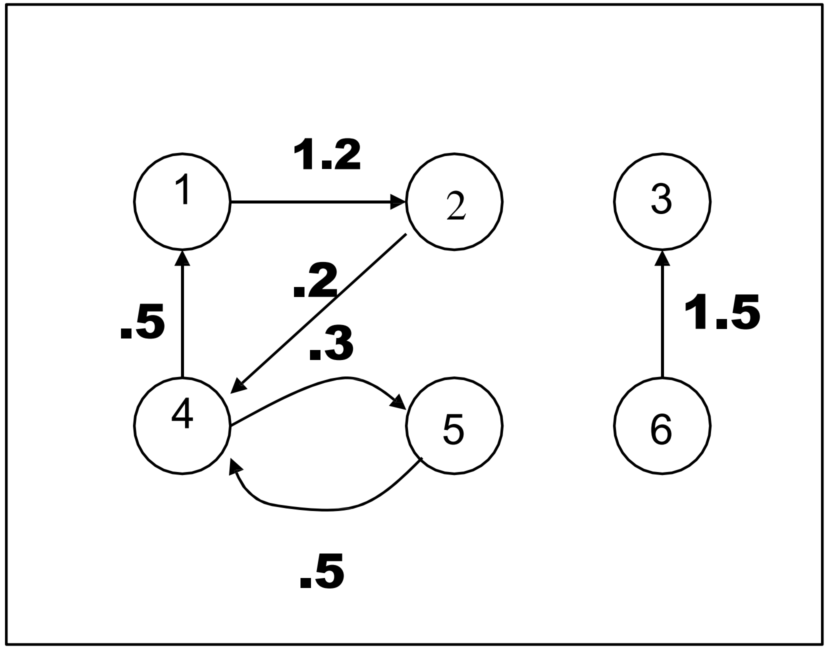
Directed graph : (digraph)

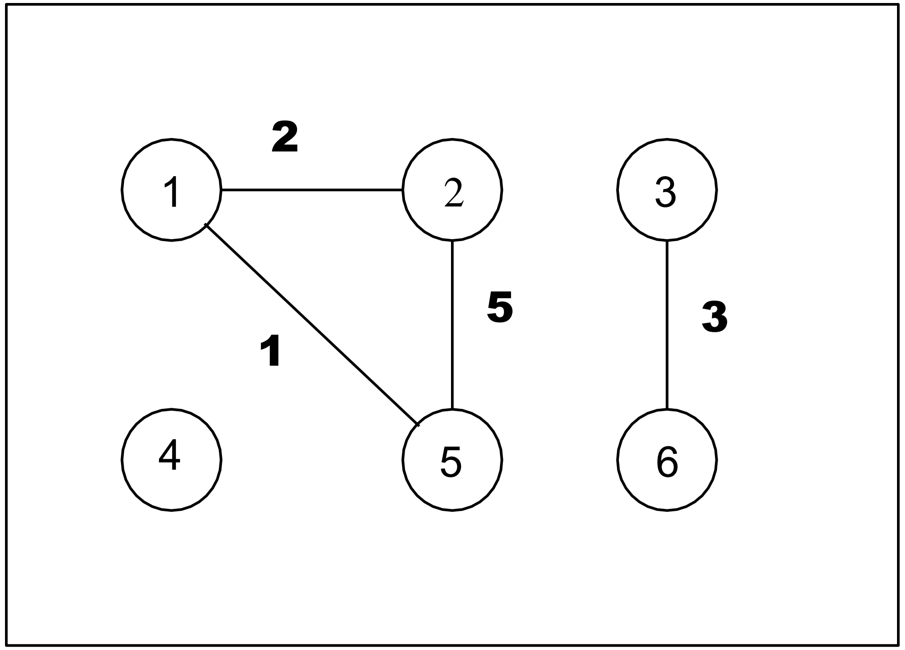
- Edges have directions



Weighted graph

- It is a graph for which each edge has an associated weight





Bipartite Graph

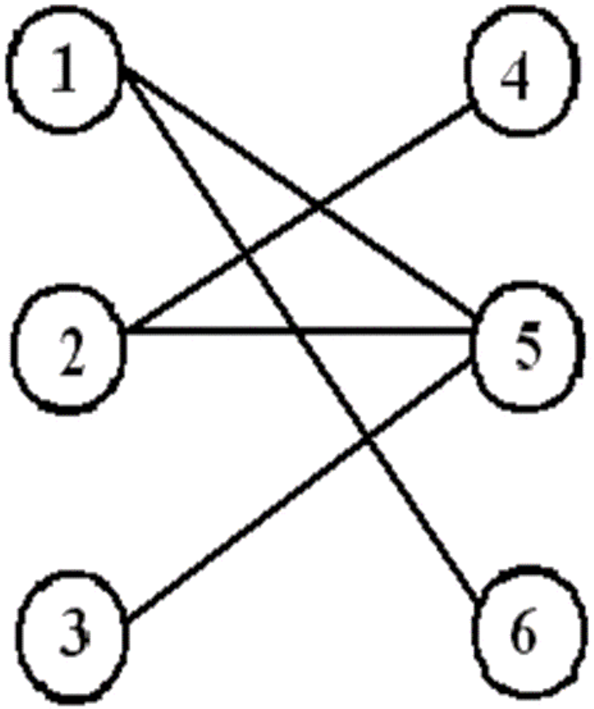
- V can be partitioned into 2 sets V_1 and V_2

such that $(u,v) \in E$ implies

either $u \in V_1$ and $v \in V_2$

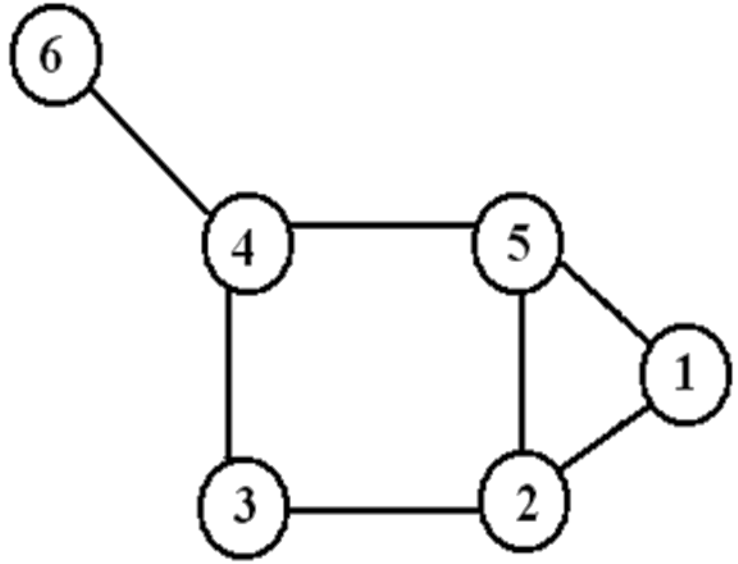
OR $v \in V_1$ and $u \in V_2$

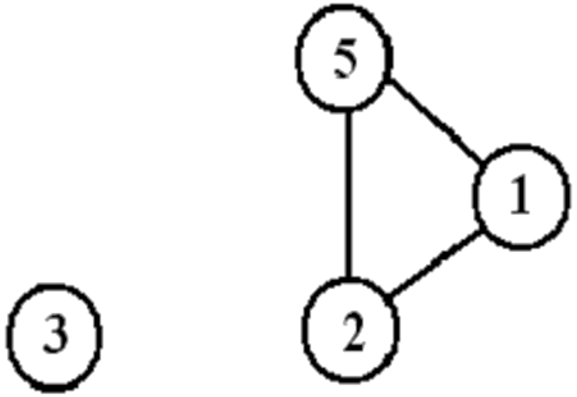
1, 5 OR 5, 1



Subgraph

- Vertex and edge sets are subsets of those of G
  - a supergraph of a graph G is a graph that contains G as a subgraph.

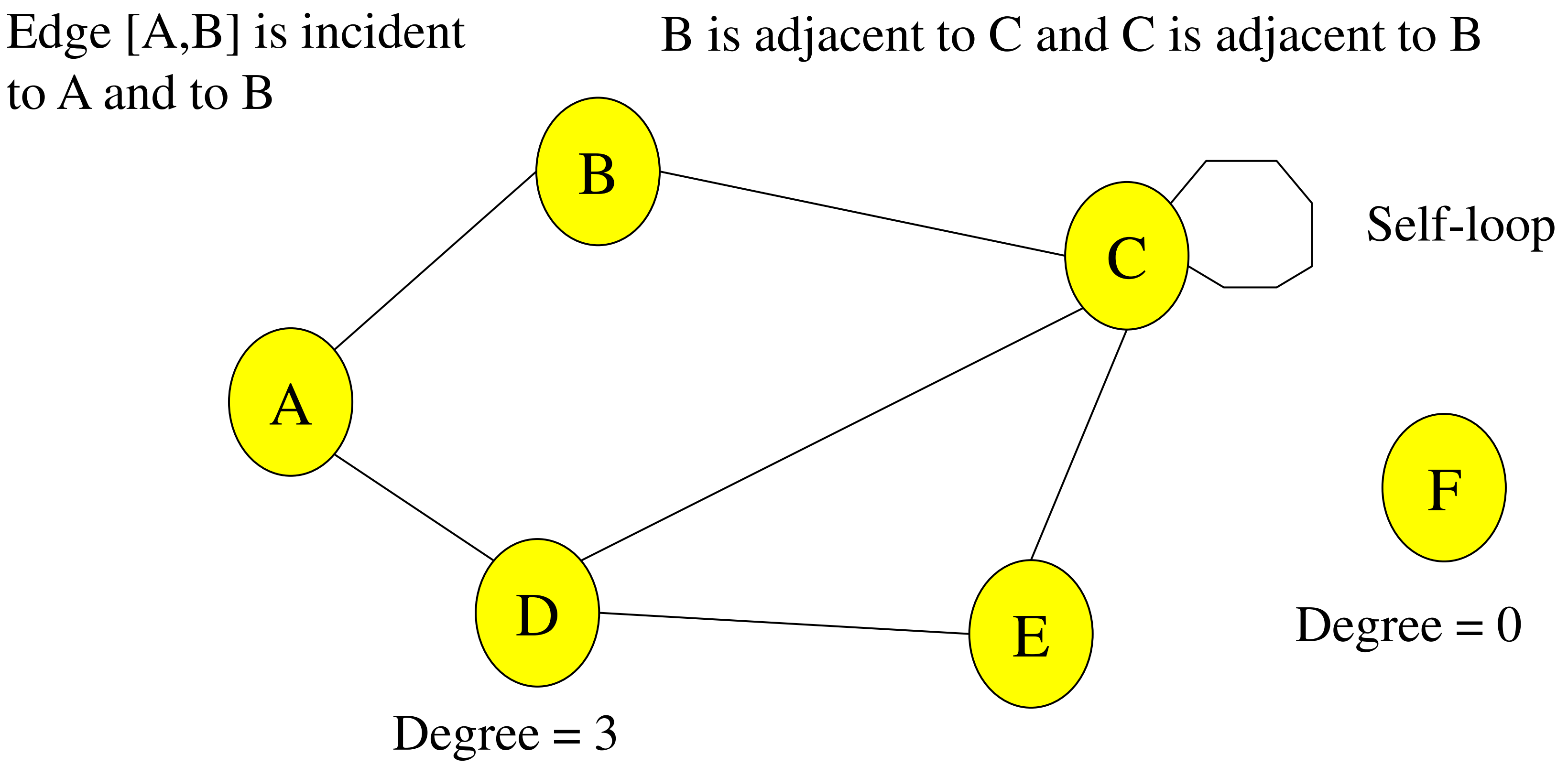




Undirected Terminology

- Two vertices u and v are adjacent in an undirected graph G if {u,v} is an edge in G
  - edge e = {u,v} is incident with vertex u and vertex v
  - Some undirected graphs allow "self loops". These will need slightly different notation, because {u,u} = {u}. Therefore, use [u,v] and [u,u] to represent the edges of such graphs.
- The degree of a vertex in an undirected graph is the number of edges incident with it
  - a self-loop counts twice (both ends count)
  - denoted with deg(v)

Undirected Graph Terminology

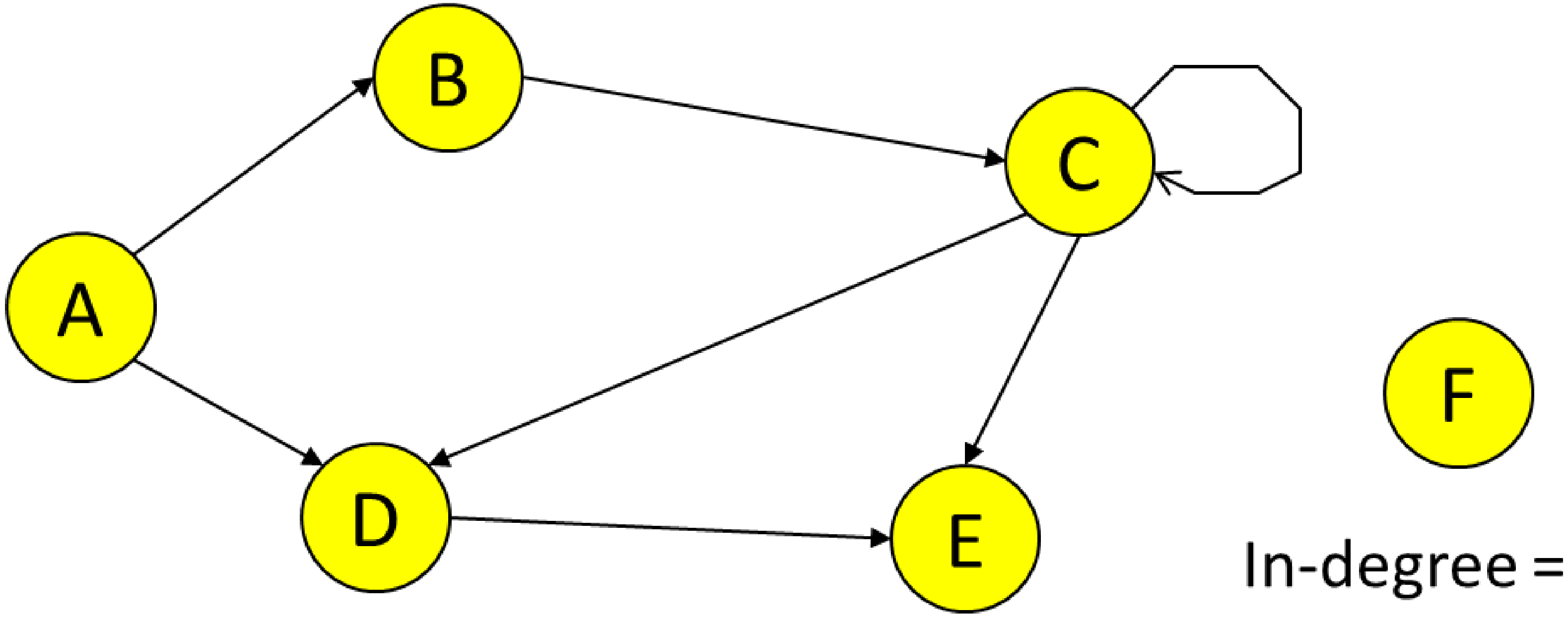


Directed Graph Terminology

- Vertex u is adjacent to vertex v in a directed graph G if (u,v) is an edge in G
  - vertex u is the initial vertex of (u,v)
- Vertex v is adjacent from vertex u
  - vertex v is the terminal (or end) vertex of (u,v)
- Degree
  - **in-degree** is the number of edges with the vertex as the terminal vertex
  - **out-degree** is the number of edges with the vertex as the initial vertex

Directed Graph Terminology

B adjacent to C and C adjacent from B

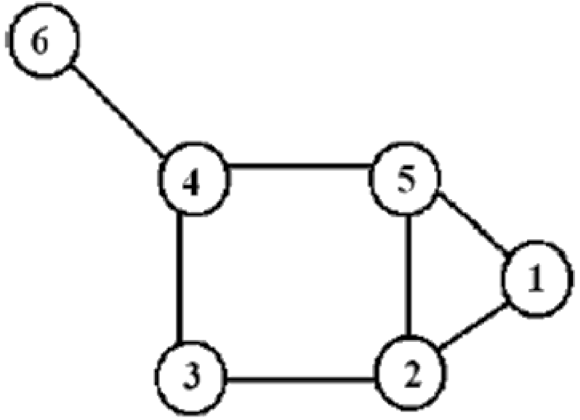


In-degree = 2
Out-degree = 1

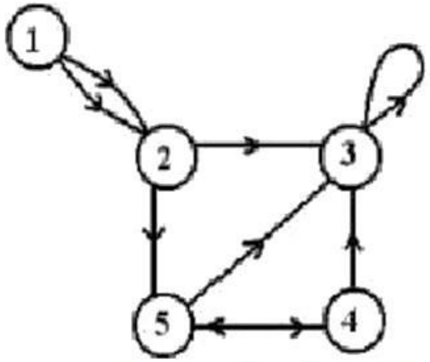
In-degree = 0
Out-degree = 0

Degree

- Number of edges incident on a node.
- Directed Graphs:
  - In-degree: Number of edges entering
  - Out-degree: Number of edges leaving
- Degree = indeg + outdeg
- A vertex with degree
  - Zero – isolated vertex
  - One – pendant vertex
  - Odd – odd vertex
  - Even – even vertex



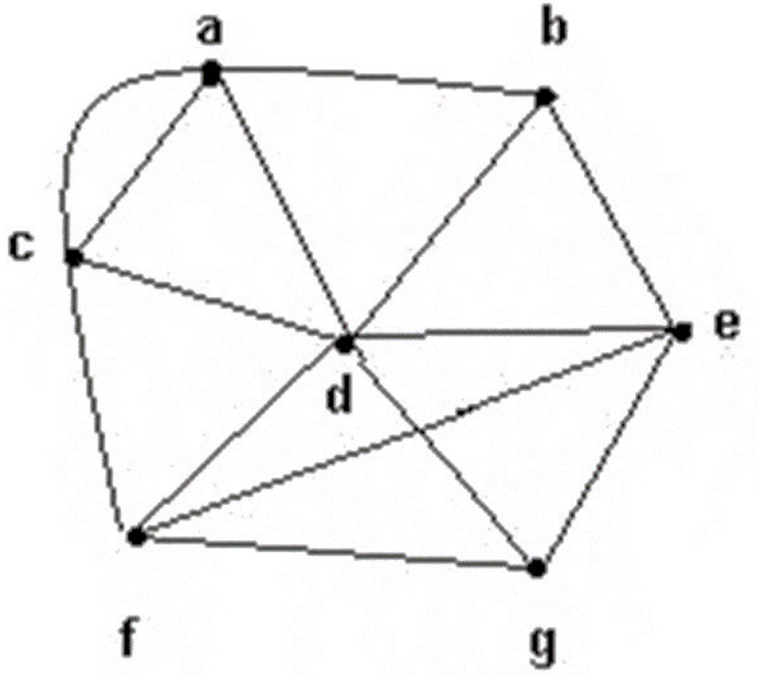
The degree of 5 is 3



outdeg(1)=2
indeg(1)=0

Degree of Vertex

- The degree of a vertex v, denoted by $\delta(v)$, is the number of edges incident on v
- Example:
- $\delta(a) = 4$, $\delta(b) = 3$,
- $\delta(c) = 4$, $\delta(d) = 6$,
- $\delta(e) = 4$, $\delta(f) = 4$,
- $\delta(g) = 3$.



Theorem 1

The sum of the degree of a graph is even, being twice the number of edges.

- If G is a graph with m edges and n vertices $v_1, v_2, \ldots, v_n$, then

$$\sum_{i=1}^{n} \delta(v_i) = 2m$$

- In particular, the sum of the degrees of all the vertices of a graph is even.

Theorem 1: Example (Undirected)

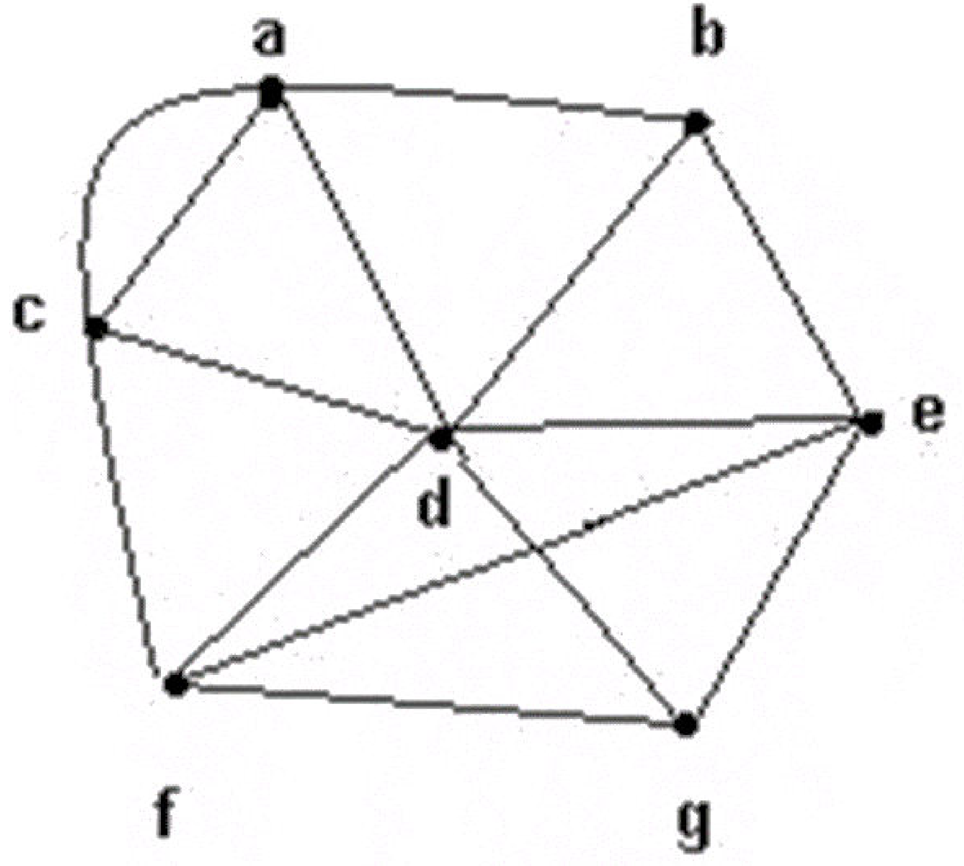
Example:

- $\delta(a) = 4$, $\delta(b) = 3$,
- $\delta(c) = 4$, $\delta(d) = 6$,
- $\delta(e) = 4$, $\delta(f) = 4$,
- $\delta(g) = 3$.

$$\sum \delta(v) = 4 + 3 + 4 + 6 + 4 + 4 + 3$$
$$= 28$$

$m = 14$

$2m = 28.$



Theorem 1: Example (Directed)

Example:

In-degree

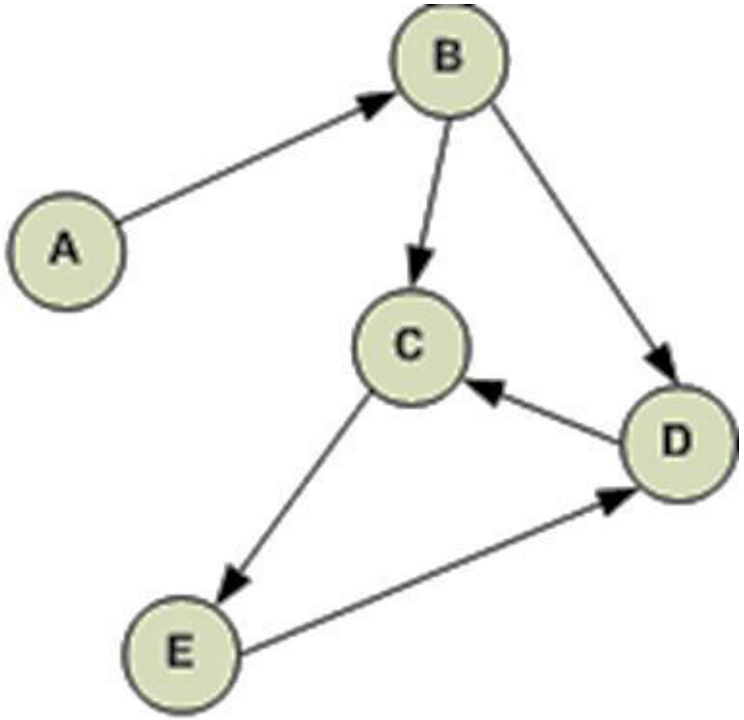
$\delta(a) = 0$, $\delta(b) = 1$, $\delta(c) = 2$,
$\delta(d) = 2$, $\delta(e) = 1$,

Out-degree

$\delta(a) = 1$, $\delta(b) = 2$, $\delta(c) = 1$,
$\delta(d) = 1$, $\delta(e) = 1$,

$$\sum \delta(v) = 1 + 3 + 3 + 3 + 2$$
$$= 12$$

$$m = 6$$
$$2m = 12.$$



Directed Graph

Theorem 1.2.

The number of vertices of odd degree in a graph is always even.

Proof:

Let G = (V, E) be a graph and d(v) be the degree of the vertex v ∈ V.

Let |E| = m.

Then $\sum_{v \in V} d(v) = 2m$ and therefore,

∑ d(vj) + ∑ d(vk) = 2m. (1.2.1)

odd degree vertices I + even degree vertices II

Since the right hand side of (1.2.1) is even, and (II) in (1.2.1) is also even, therefore (I) in (1.2.1) is even.

Hence, ∑d(vj) = even number.

odd degree vertices

This is only possible when the number of vertices with odd degree is even.

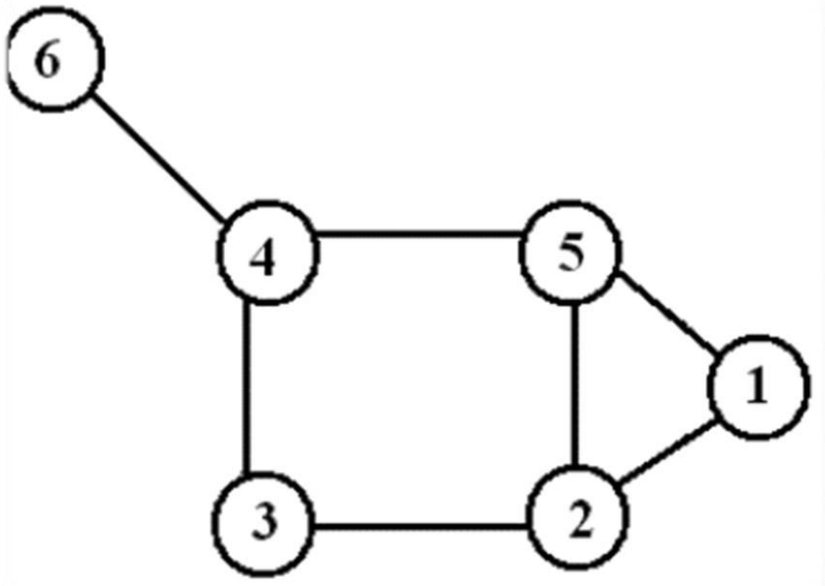
Theorem 2

Example

- No-of Vertices = 6 (even)
- Degree: 2 + 3 + 2 + 3 + 3 + 1 = 14

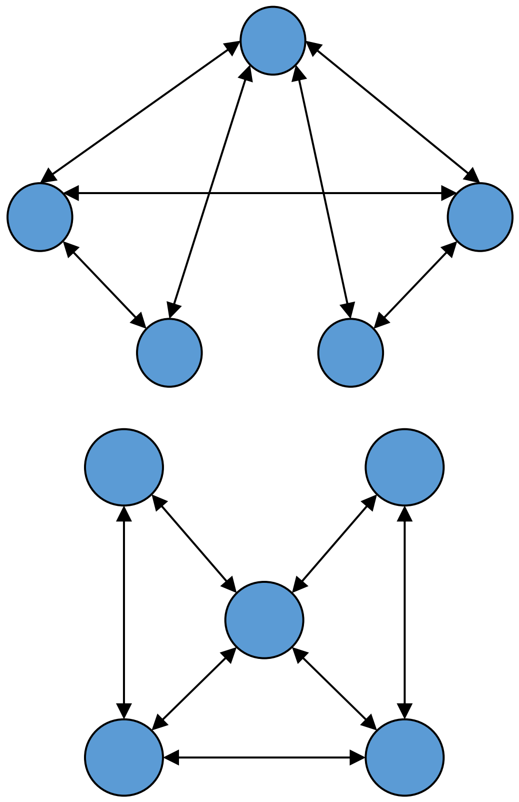
$\sum_{v \in V} d(v) = 2m$ and therefore,

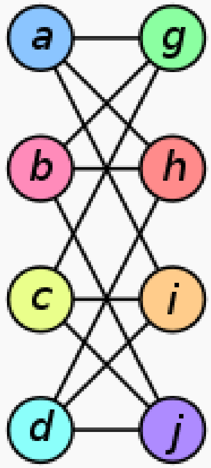
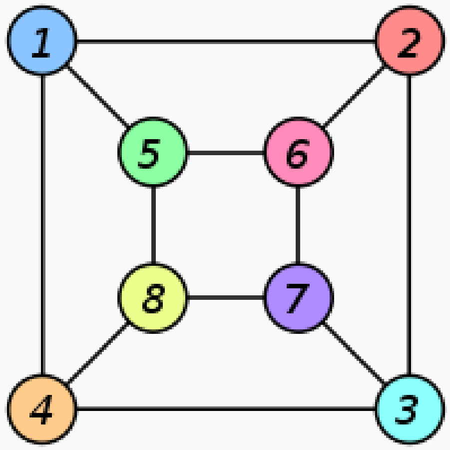
- = 2 * m
- = 2 * 14 = 28.
- Odd degree vertex = V1, V3, V5 = 2 + 2 + 3 = 7
- Even degree vertex = V2, V4, V6 = 3 + 3 + 1 = 7



Isomorphism

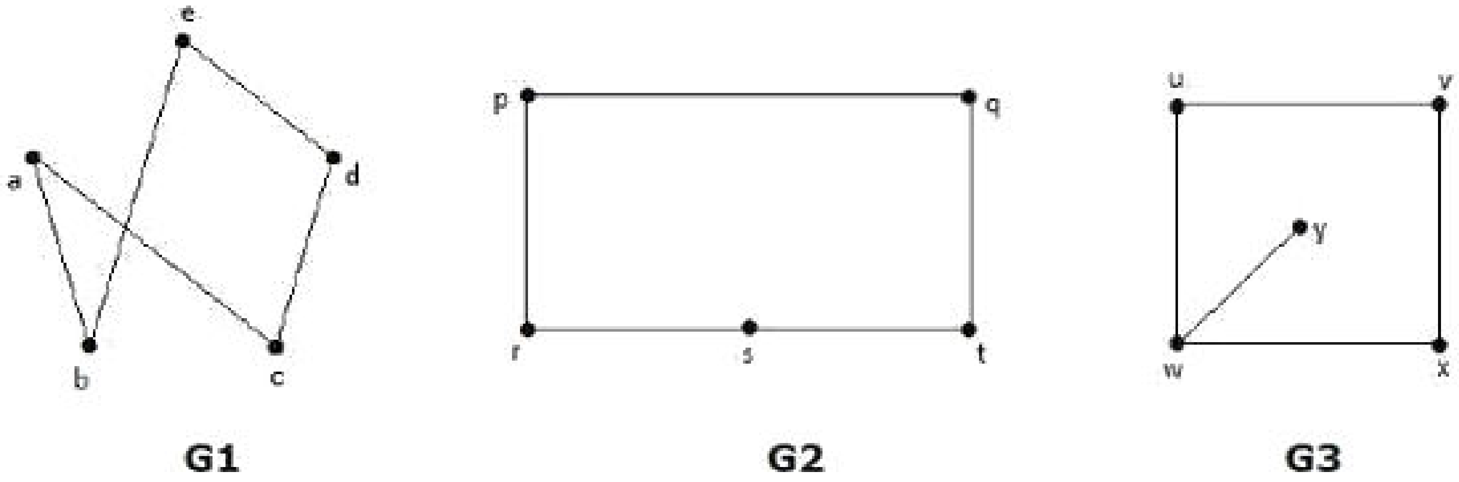
A graph can exist in different forms having the same number of vertices, edges, and also the same edge connectivity. Such graphs are called isomorphic graphs.



<table>
<tr><th>Graph G</th><th>Graph H</th><th>An isomorphism between G and H</th></tr>
<tr><td></td><td></td><td>f(a) = 1
f(b) = 6
f(c) = 8
f(d) = 3
f(g) = 5
f(h) = 2
f(i) = 4
f(j) = 7</td></tr>
</table>

Isomorphic Graphs

- Two graphs G1 and G2 are said to be isomorphic if
  - ➔The same number of vertices,
  - ➔The same number of edges, and
  - ➔An equal number of vertices with a given degree.



- In the graph G_3, vertex 'w' has only degree 3, whereas all the other graph vertices has degree 2. Hence G3 not isomorphic to G_1 or G_2.

Theorem 1.3

The relation isomorphism in graphs is an equivalence relation.

Proof:

The relation of isomorphism between graphs is reflexive because of the trivial automorphisms.

Let $\theta = (f , g)$ be an isomorphism of a graph G onto a graph H, so $G \cong H$. Then there is an inverse isomorphism

$\theta^{-1} = (f^{-1} , g^{-1})$ of H onto G. So $H \cong G$. Therefore $\cong$ is symmetric.

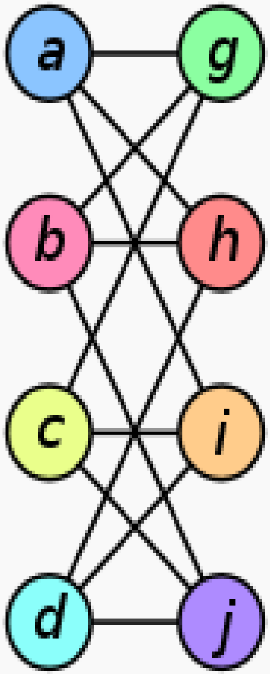
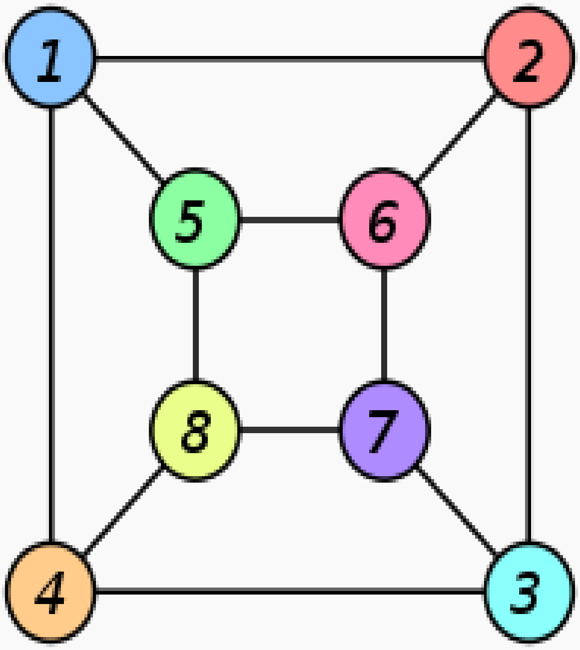
Now, let $\theta = (f , g)$ be an isomorphism of G onto H, and $\phi\theta = (f1 f , g1g)$ of G onto K. Here f1 f is a mapping obtained by applying first f and then f1. Similarly, $\phi\ \theta$ is the isomorphism obtained applying first θ and then ϕ. Thus $\cong$ is transitive.

Hence the relation isomorphism is an equivalence relation

Theorem 1.3

Example

$G(a) = 3$ $\quad$ $G(1) = 3$

Graph G	Graph H	An isomorphism between G and H
		$f(a) = 1$ $f(b) = 6$ $f(c) = 8$ $f(d) = 3$ $f(g) = 5$ $f(h) = 2$ $f(i) = 4$ $f(j) = 7$

Theorem 1.4

Let G and H be graphs and let f be a one–one mapping V(G) onto V(H) such that two distinct vertices x and y of G are adjacent if and only if the corresponding vertices f x and f y of H are adjacent in H. Then there is a uniquely determined one–one mapping g of E(G) onto E(H) such that (f , g) is an isomorphism of G onto H.

Proof:

Let e be any edge of G, having distinct ends x and y. By hypothesis, there is a uniquely determined edge e' of H whose ends are f x and f y. We define a one–one mapping g by the rule ge = e', for each edge e of G. It is then clear that (f, g) is an isomorphism of G onto H.

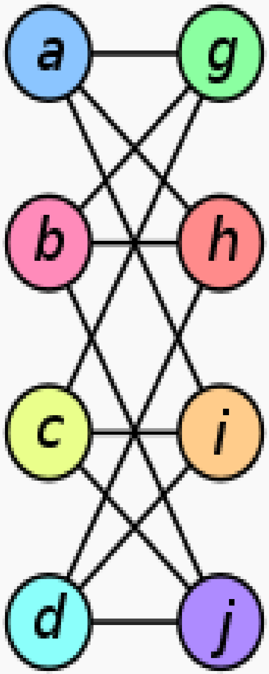
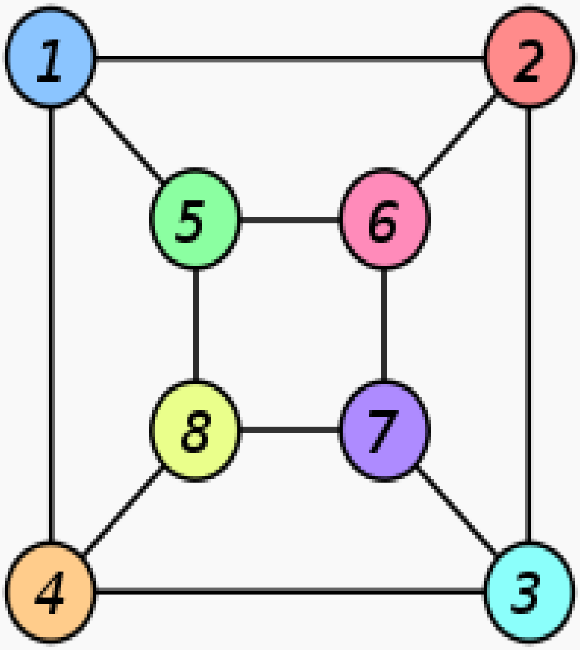
Conversely, let g be a mapping such that (f, g) is an isomorphism of G onto H. Then for each edge e of G, there is an edge e' of H such that ge = e' .

An isomorphism of a graph G onto a graph H is defined as a one–one mapping of V(G) onto V(H) that preserves adjacency.

Theorem 1.4

Example

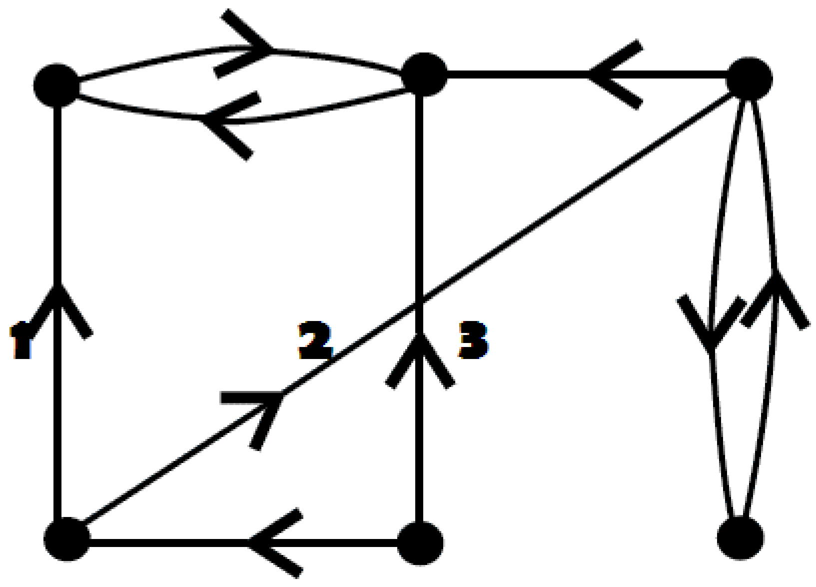
Adjacent

Graph G	Graph H	An isomorphism between G and H
		$f(a) = 1$ $f(b) = 6$ $f(c) = 8$ $f(d) = 3$ $f(g) = 5$ $f(h) = 2$ $f(i) = 4$ $f(j) = 7$

Types of Graph

Simple directed graph or simple digraph

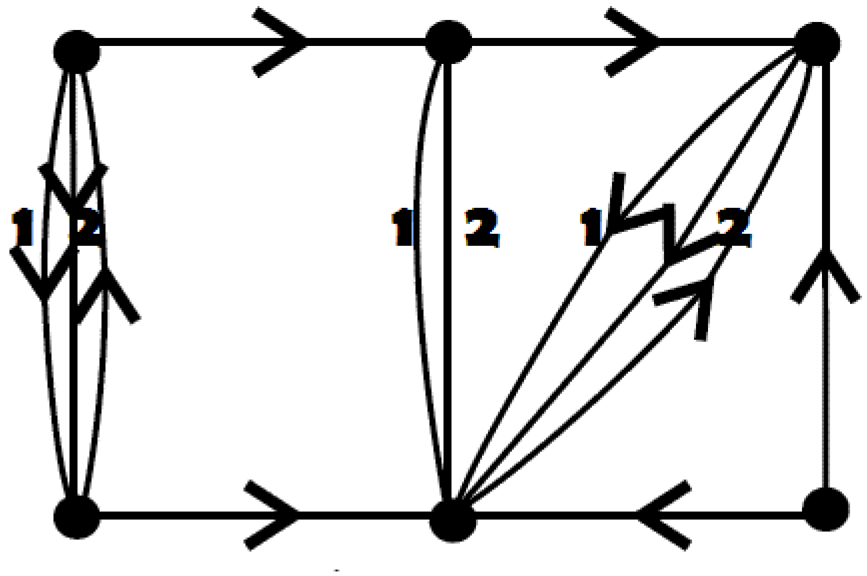
- A simple digraph (or simply digraph) D is a pair (V, A), where V is a nonempty set of vertices and A is a subset of V whose elements are called ***arcs*** of D.



Types of Graph

Multidigraph

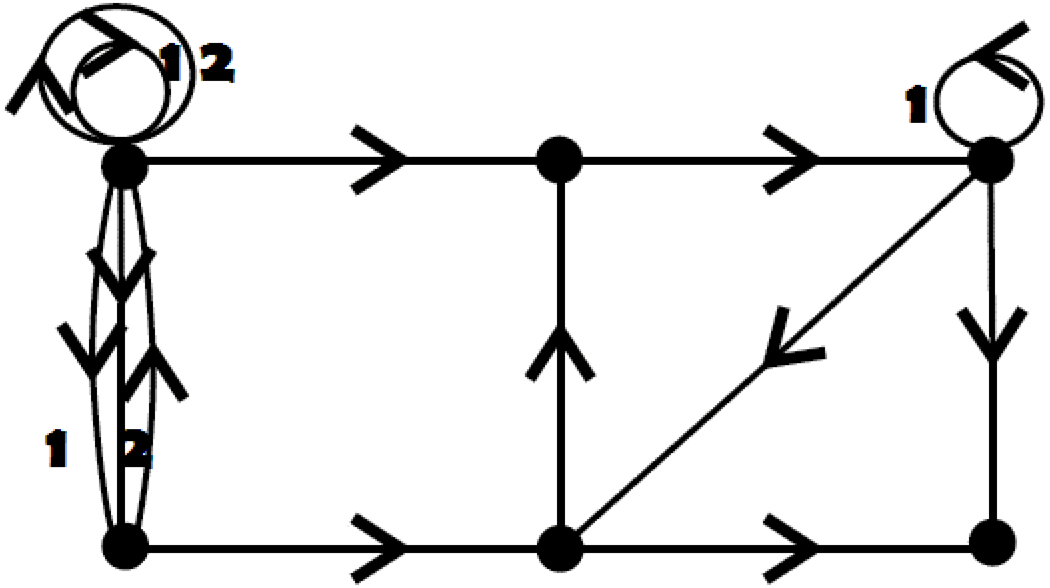
- A multidigraph D is a pair (V, A), where V is a nonempty set of vertices, and A is a multiset of arcs of V.
- The number of times an arc occurs in D is called its multiplicity and arcs with multiplicity greater than one are called multiple arcs of D.



Types of Graph

General digraph

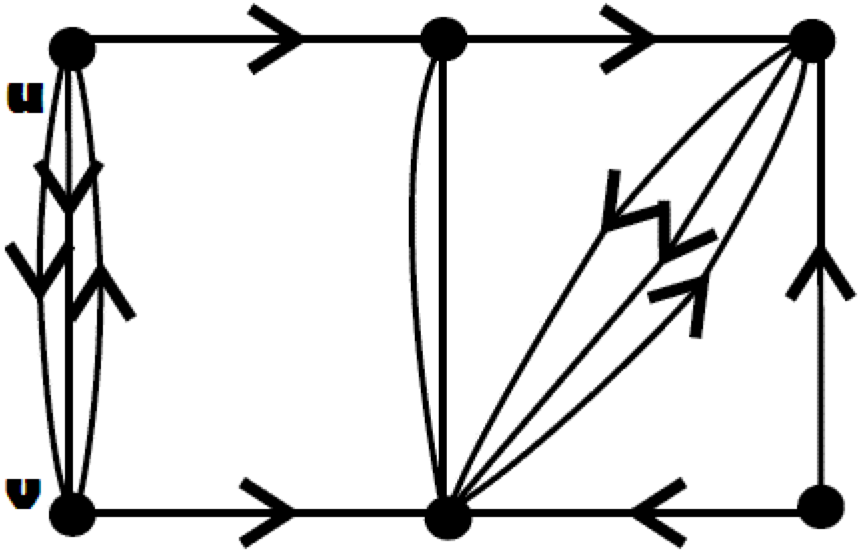
- A general digraph D is a pair (V, A), where V is a nonempty set of vertices, and A is a multiset of arcs, being a multisubset V ×V.
- An arc of the form uu is called a loop of D and arcs which are not loops are called proper arcs of D.
- The number of times an arc occurs is called its multiplicity.
- A loop with multiplicity greater than one is called a multiple loop.



Types of Graph

General digraph

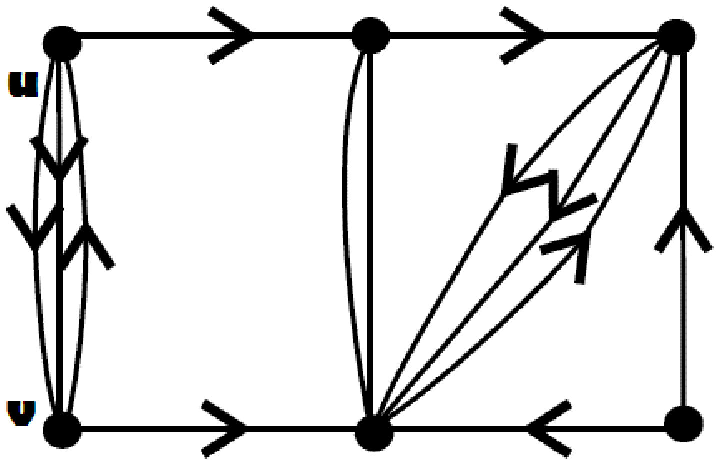
- An arc (u, v) ϵ A of a digraph is denoted by uv, implying that it is directed from u to v, u being the initial vertex and v the terminal vertex. Clearly, a digraph is an irreflexive binary relation on V.



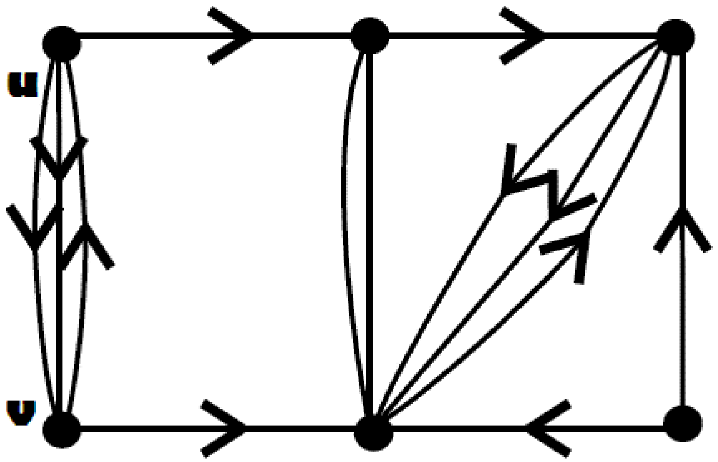
Types of Graph

General digraph

- If D(V, A) is a digraph, the graph G(V, E), where uv ϵ E whenever uv or vu or both are in A, is called the **underlying graph** of D (also called the **covering graph** C(D) of D).



D(V, A)



G(V, E)

Types of Graph

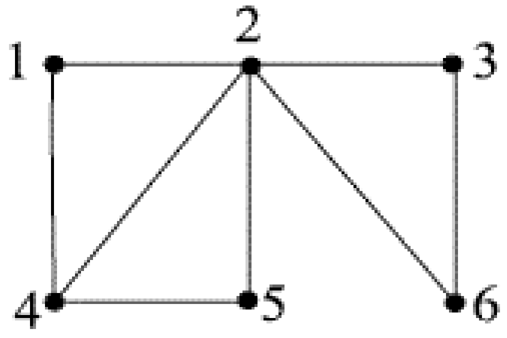
Mixed graph

- A mixed graph G{V, A∪E} consists of a nonempty set V of vertices, a set A of arcs (A ⊆ V), and a set E of edges (E ⊆ V), such that if uv ϵ E then neither uv, nor vu is in A.
- We represent a general, multi and simple graph by g-graph, m-graph and s-graph respectively.

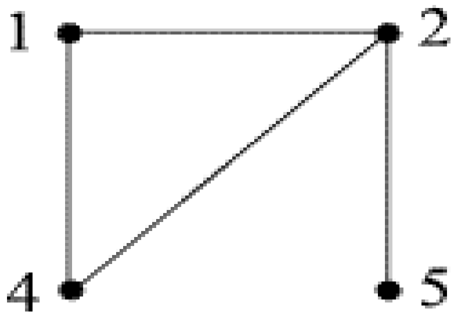
Types of Graph

Subgraph

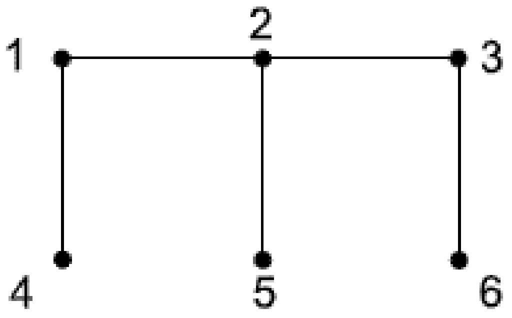
- A subgraph of a graph G(V, E) is a graph H(U, F) with U ⊆ V and F ⊆ E.
- We denote it by H < G (G is also called the super graph of H.) If U = V then H is called the spanning subgraph of G, and is denoted by H ≤ G. Here G is called the spanning super graph of H and is denoted by G ≥ H.



Graph G



Subgraph H

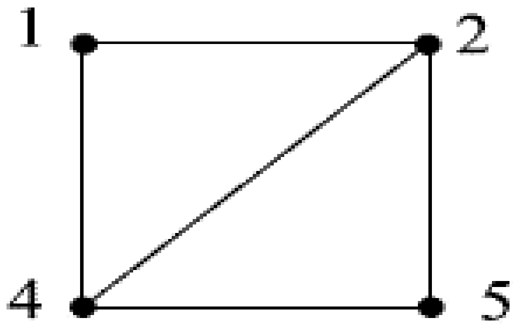


Spanning subgraph

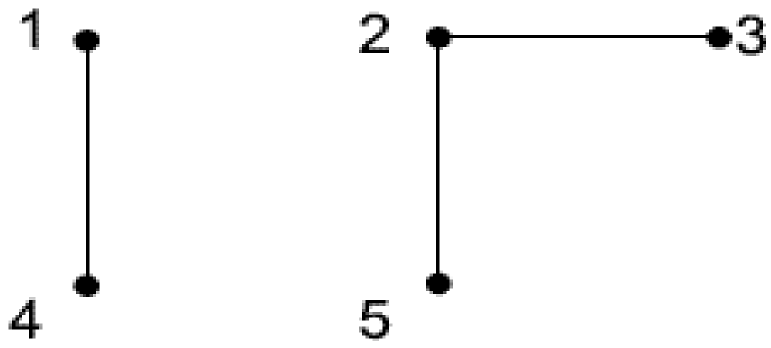
Types of Graph

Vertex induced subgraph/Edge induced subgraph

- If F consists of all those edges of G joining pairs of vertices of U, then H is called the vertex induced subgraph of G and is denoted by H =< U >.
- If F ⊆ E, and U is the set of end vertices of the edges of F, then H(U, F) is called an edge induced subgraph of G and is denoted by H =< F >.



Vertex induced subgraph
< U > = < 1, 2, 4, 5 >

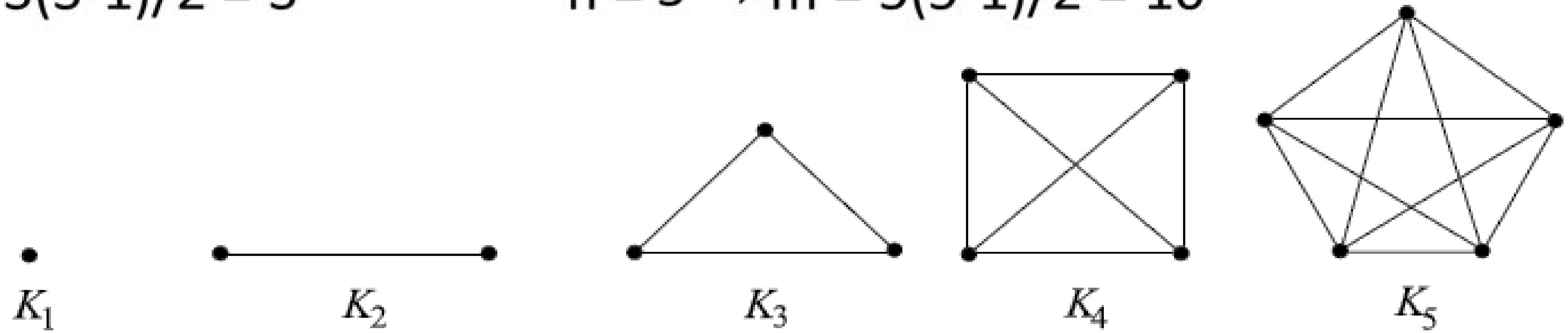


Edge induced subgraph
< *F* > = < 14, 25, 23 >

Types of Graph

Complete Graph

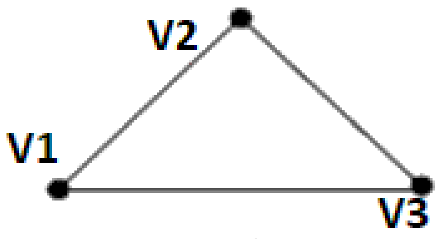
- A graph of order n with all possible edges $m = n(n-1)/2$ is called a complete graph of order n and is denoted by K_n.
- A graph of order n with no edges is called an empty graph and is denoted by $\overline{K_n}$.
- Each graph of order n is clearly a spanning subgraph of K_n.
- $n = 1 \rightarrow m = 1(1-1)/2 = 0$
- $n = 2 \rightarrow m = 2(2-1)/2 = 1$ $n = 4 \rightarrow m = 4(4-1)/2 = 6$
- $n = 3 \rightarrow m = 3(3-1)/2 = 3$ $n = 5 \rightarrow m = 5(5-1)/2 = 10$



Types of Graph

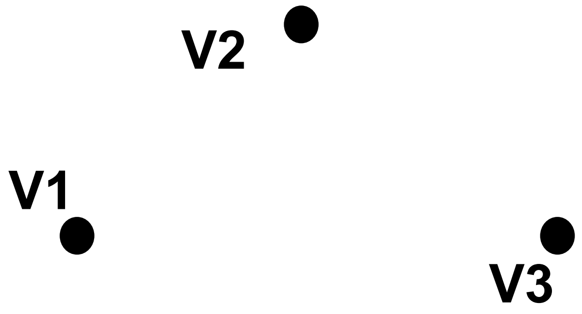
r-partite graph

- A graph $G(V, E)$ is said to be *r-partite* (where r is a positive integer) if its vertex set can be partitioned into disjoint sets $V_1, V_2, \ldots, V_r$ with $V = V_1 \cup V_2 \cup \ldots \cup V_r$ such that uv is an edge of G if u is in some V_i and v in some V_j, $i \neq j$.
- That is, every one of the induced subgraphs $< V_i >$ is an empty graph. We denote r-partite graph by $G(V_1, V_2, \ldots, V_r, E)$.



Complete r-partite graph

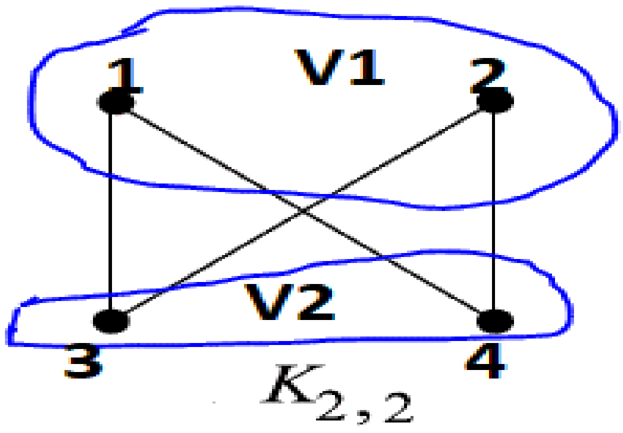
- If an *r-partite* graph has all possible edges, that is, $uv \in E$, for every $u \in V_i$ and every $v \in V_j$, for all i, j, i $\neq$ j, then it is called a ***complete r-partite*** graph.
- If $|V_i| = n_i$, we denote it by $k_{n1, n2, \ldots, nr}$.



Types of Graph

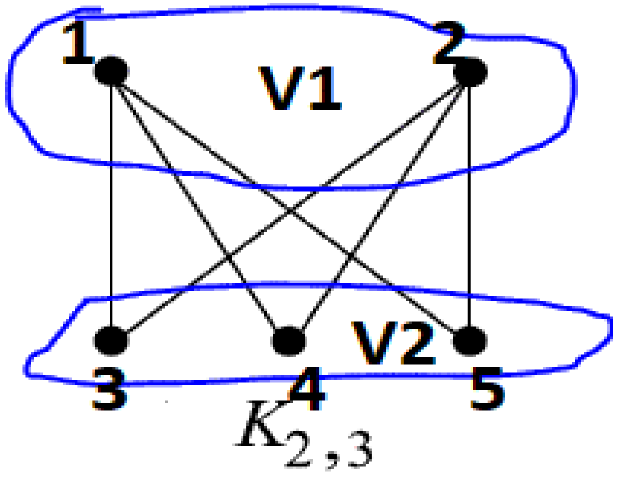
Bipartite graph

- A graph ***G(V, E)*** is said to be bipartite, or 2-partite, if its vertex set can be partitioned into two different sets V_1 and V_2 with $V = V_1 \cup V_2$, such that $uv \in E$ if $u \in V_1$ and $v \in V_2$.



- $V = V_1 \cup V_2$
- $V_1 = (1, 2)$ $\quad\quad$ $V_2 = (3, 4)$
- $V = V_1 \cup V_2$ $(1, 2, 3, 4)$
- $E = (1, 3)\ (1, 4)\ (2, 3)\ (2, 4)$

u v



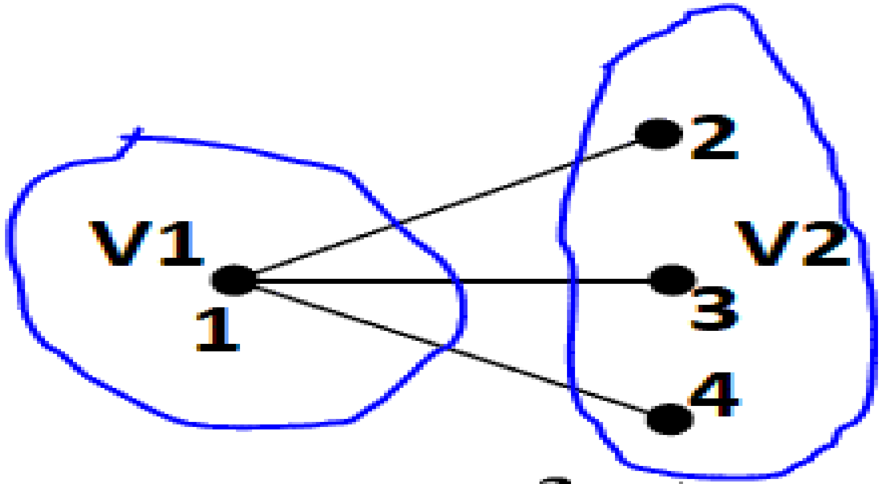
- $V = V_1 \cup V_2$
- $V_1 = (1, 2)$ $\quad\quad$ $V_2 = (3, 4, 5)$
- $V = V_1 \cup V_2$ $(1, 2, 3, 4, 5)$
- $E = (1, 3)\ (1, 4)\ (1, 5)\ (2, 3)\ (2, 4)\ (2, 5)$

u v u v

Types of Graph

Complete Bipartite graph

- The bipartite graph is said to be complete if $\boldsymbol{uv} \in \boldsymbol{E}$, for every $\boldsymbol{u} \in \boldsymbol{V_1}$ and every $\boldsymbol{v} \in \boldsymbol{V_2}$.
- When $|V_1| = n_1$, $|V_2| = n_2$, we denote the complete bipartite graph by $K_{n1,\, n2}$.
- The complete bipartite graph $K_{1,n}$ is called an *n-star* or *n-claw*.
- $V_1 \rightarrow E = (1, 2)\ (1, 3)\ (1, 4) \quad u = (1)$
- $V_2 \rightarrow E = (1, 2)\ (1, 3)\ (1, 4) \quad v = (2, 3, 4)$



3-star

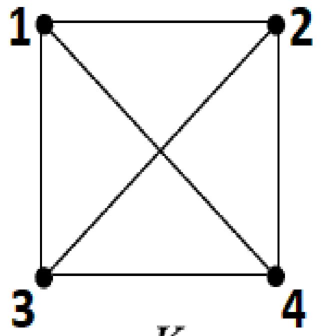
Complement of a Graph

- The complement $\overline{G}(V, \overline{E})$ of a graph $G(V, E)$ is the graph having the same vertex set as G, and its edge set $\overline{E}$ is the complement of E in $V_{(2)}$, that is, uv is an edge of $\overline{G}$ if and only if uv is not an edge of G.
- $G(V, E)$
- $V = (1, 2, 3, 4) = 4$
- $E = (1, 2)\ (1, 3)\ (1, 4)$

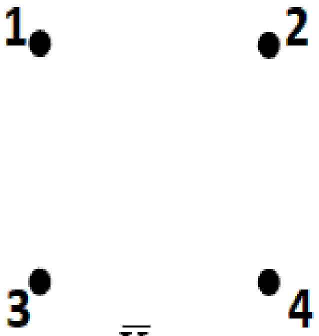
  $(2, 3)\ (2, 4)\ (3, 4) = 6$
- $V = (1, 2, 3, 4, 5) = 5$
- $E = (1, 3)\ (1, 4)\ (1, 5)$

  $(2, 3)\ (2, 4)\ (2, 5) = 6$

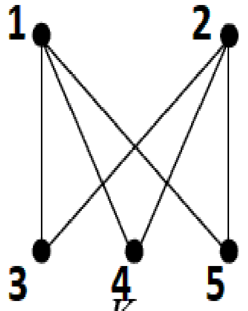
$\overline{E} = (1, 2)\ (3, 4)\ (4, 5)\ (3, 5) = 4$



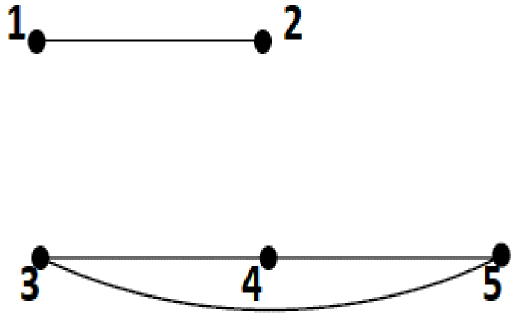
K_4



$\overline{K}_4$



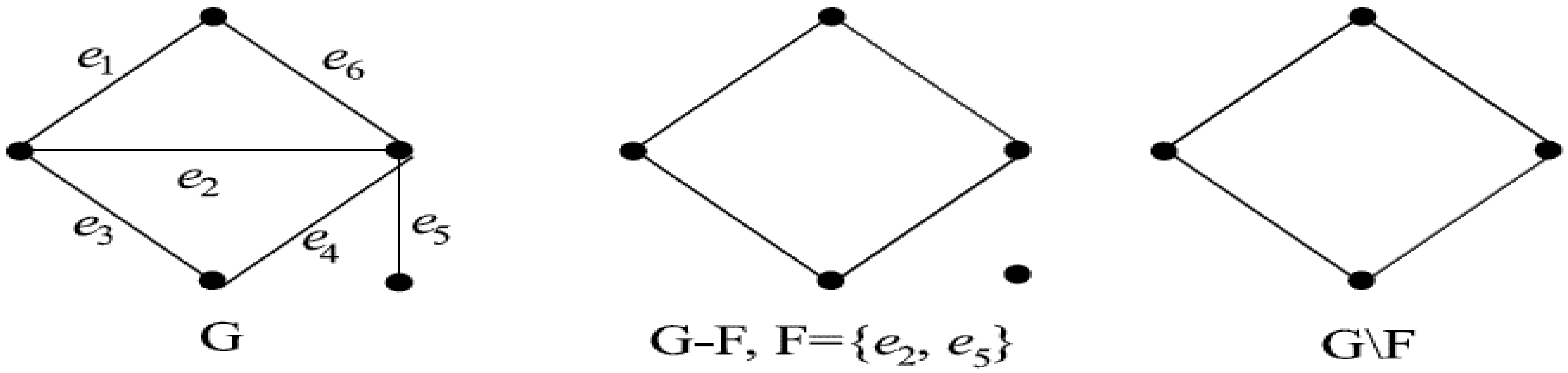
$K_{2,3}$



$\overline{K}_{2,3}$

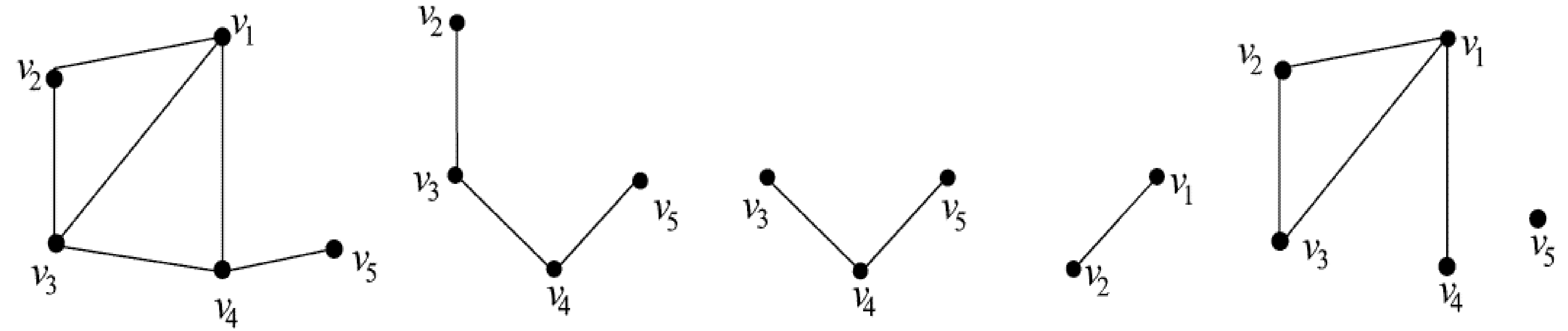
Removal of Edges

- ***G(V, E)*** be a graph and let $F \subseteq E$. The graph ***H(V, E − F)*** with vertex set ***V*** and edge set ***E−F*** is said to be obtained from ***G*** by removing the edges in ***F***.
- It is denoted by ***G−F***. If F consists of a single edge ***e*** of ***G***, the graph obtained by removing ***e*** is denoted by ***G−e***.
- ***G−F*** may contain isolated vertices which are not isolated vertices of ***G***. The graph obtained by removing these newly created isolated vertices from ***G−F*** is denoted by ***G\F***.
- Similarly, the graph obtained by removing isolated vertices from ***G−e*** is denoted by ***G\ e***.



Removal of Vertices

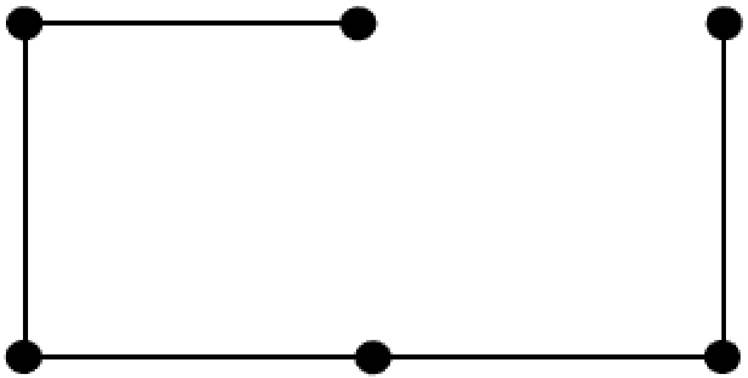
- Let $G(V, E)$ be a graph and let $v \in V$. Let E_v be the set of all edges of G incident with v. The graph $H(V - \{v\}, E - E_v)$ is said to be obtained from G by the removal of the vertex v and is denoted by $G-v$.
- If U is a subset of V, the graph obtained by removing the vertices of G which are in U is denoted by $G-U$.
- If H is a subgraph of G, we denote $G-V(H)$ by $G-H$, and $G-E(H)$ by $\overline{H}\,(G)$. Here $\overline{H}(G)$ is called the relative complement of H in G.



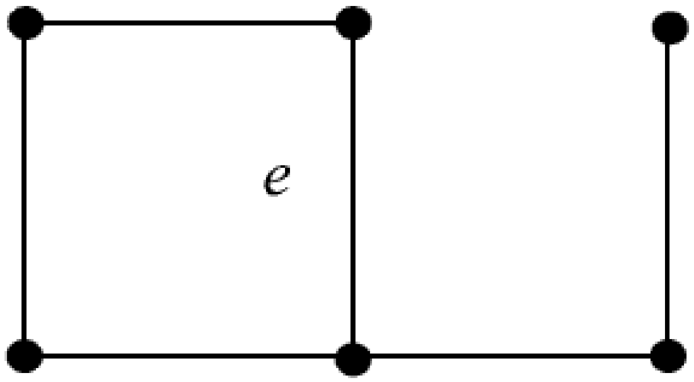
G G-v1 H (Subgraph of G) G-v(H) G-E(H)

Addition of Edges

- Let ***G(V, E)*** be a graph and let ***f*** be an edge of $\overline{G}$.
- The graph ***H(V, E*** ∪ ***{ f })*** is said to be obtained from G by the addition of the edge ***f*** and is denoted by ***G + f*** .
- If ***F*** is a subset of edges of $\overline{G}$, the graph obtained from ***G*** by adding the edges of ***F*** is denoted by ***G+F***.



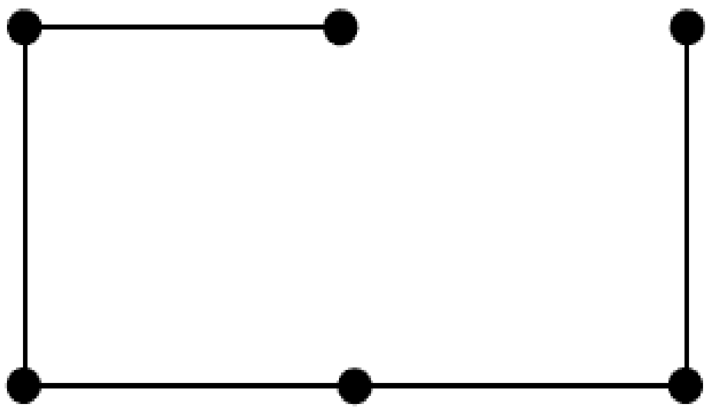
Graph G



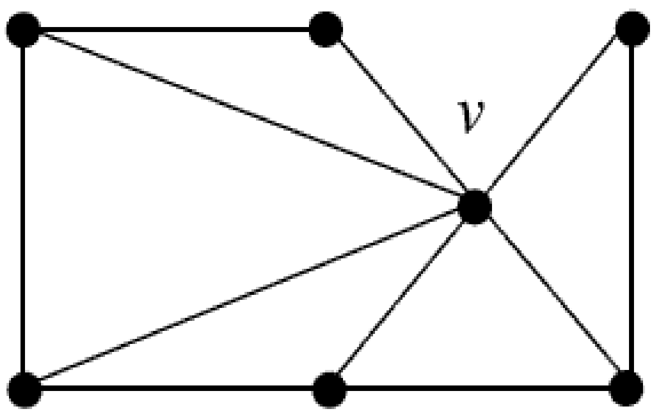
$G + e$

Addition of Vertices

- Let ***G(V, E)*** be a graph and let ***v*** $\notin$ ***V***.
- The graph ***H*** with vertex set ***V*** ∪ ***{v}*** and edge set ***E*** ∪ ***{uv, for all u*** ∈ ***V}*** obtained from ***G*** by adding a vertex ***v***, is denoted by ***G+v***.
- Thus ***G+v*** is obtained from ***G*** by adding a new vertex and joining it to all vertices of ***G***.



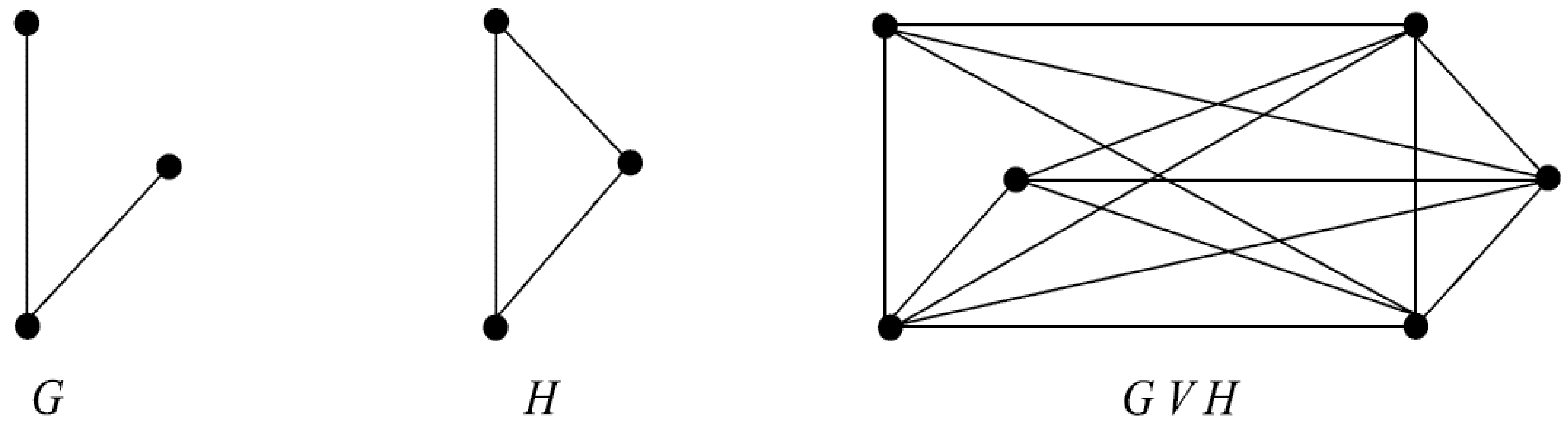
Graph G



$G + v$

Join of Graphs

- Let ***G(V, E)*** and ***H(U, F)*** be two graphs with disjoint vertex sets $(V \cap U = \Phi)$.
- The join of ***G*** and ***H*** denoted by ***G V H*** is the graph with vertex set $V \cup U$ and edge set $E \sqcup F \sqcup [V, U]$.
- So the join is obtained from ***G*** and ***H*** by joining every vertex of ***G*** to each vertex of ***H*** by an edge.
- Clearly, $G+v = G \vee K_1 = G \vee \overline{K_1}$.



Graphs Properties

- **Parametric property:** A property ***P*** is called a parametric property of a graph if for a graph G having property ***P***, every graph isomorphic to ***G*** also has property ***P***. A graph with property ***P*** is denoted as ***P-graph***.
- **P-critical:** A graph ***G*** is said to be P-critical if ***G*** is a P-graph and for every $v \in V$, $G - v$ is not a P-graph.
- **Hereditary property:** A property ***P*** is said to be a hereditary property of a graph ***G*** if a graph ***G*** has the property ***P***, then every subgraph of ***G*** also has the property ***P***. It is called an induced-hereditary property if every induced subgraph of ***G*** also has the property.
- **Monotone property:** A property ***P*** is said to be a monotone property of a graph ***G***, if G has the property ***P***, then for every $e \in E(\overline{G})$, $G+e$ also has the property P.

Paths, Cycles and Components

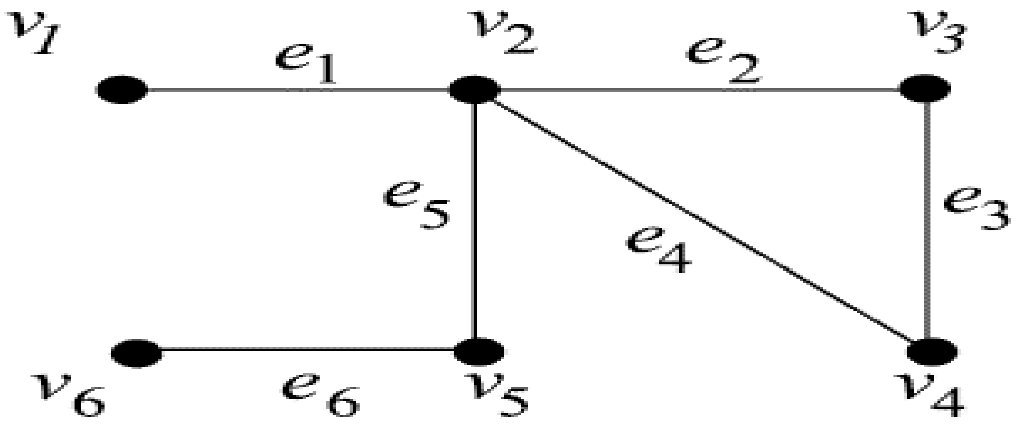
- **Walks:** An alternating sequence of vertices and edges, beginning and ending with vertices such that no edge is traversed or covered more than once is called a walk.

  Walk - $v_1e_1v_2e_2v_3e_3v_4e_4v_2e_5v_5e_6v_6$

- **Paths:** A path is an open walk in which no vertex (and therefore no edge) is repeated.

  Path1 - $v_1e_1v_2e_2v_3e_3v_4$

  Path2 - $v_1e_1v_2e_5v_5e_6v_6$



Graph G

Path

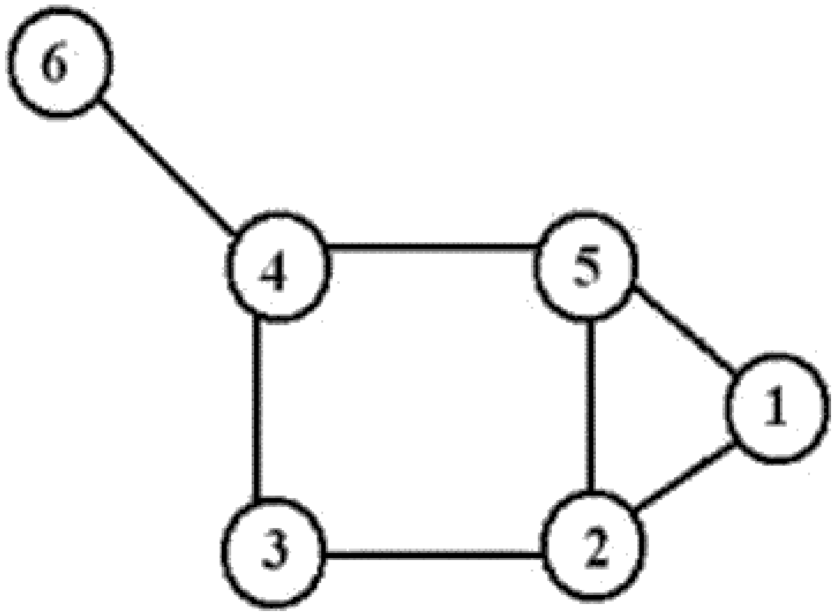
- **Walk**: A walk of length k in a graph is a succession of k (not necessarily different) edges of the form.

  Ex: 1,2,5,2,3,4,6

- **Path**: A path is a walk in which all the edges and all the nodes are different

  Ex: 1,2,3,4,6

  Ex: 1,5,4,6



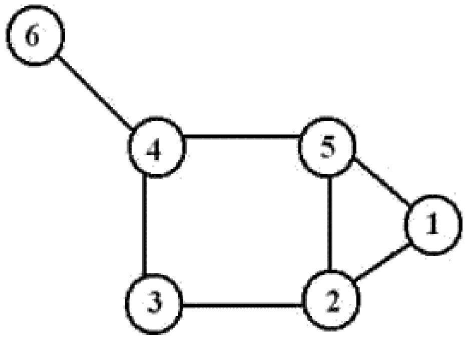
Cycle and Tree

- **Cycle**: A cycle is a closed path in which all the edges are different.

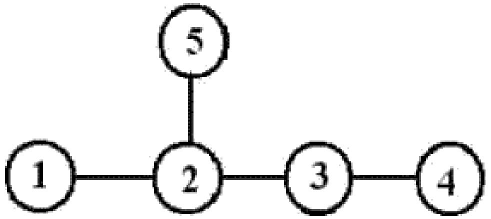
  Ex: 1,2,5,1 (3-Cycle)

  1,2,3,4,5,1

  2,3,4,5,2



- **Tree**: Connected Acyclic Graph and Two nodes have exactly one path between them



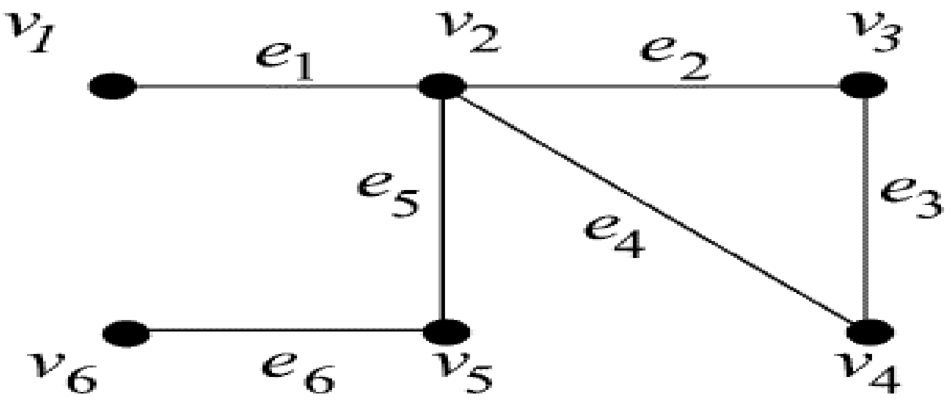
Paths, Cycles and Components

- **Cycle:** A closed walk in which no vertex (and edge) is repeated is called a cycle.

  Cycle - $v_2e_2v_3e_3v_4e_4v_2$

- **Component of a graph:** A maximal connected subgraph of a graph G is called a component of G. A component which is K_1 is called a trivial component. The number of components of a graph G is denoted by k(G).

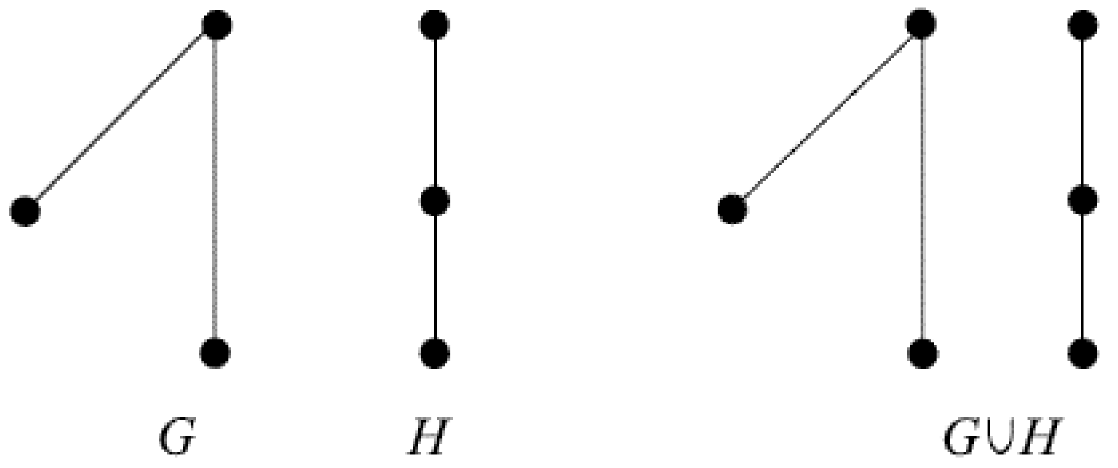
- A component of G with an odd (even) number of vertices is called an odd (even) component of G



Graph G

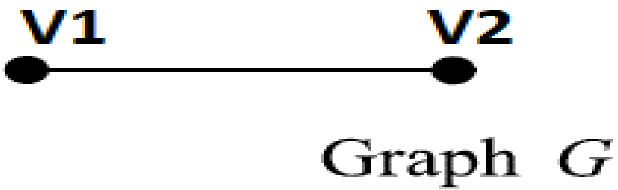
Union of Graph

- Let G(V, E) and H(U, F) be two graphs with V ∩ U = Φ.
- The union of G and H, denoted by G ∪ H, is the graph with vertex set V ∪ U and edge set E ∪ F.
- Clearly, when G and H are connected graphs, G ∪ H is a disconnected graph whose components are G and H.



Theorem 1.5

A graph G is disconnected if and only if its vertex set V can be partitioned into two nonempty, disjoint subsets V_1 and V_2, such that there exists no edge in G whose one end vertex is in V_1 and the other in V_2.



Proof:

Let G be a graph whose vertex set can be partitioned into two nonempty disjoint subsets V_1 and V_2, so that no edge of G has one end in V_1 and the other in V_2.

Let v_1 and v_2 be any two vertices of G such that $v_1 \in V_1$ and $v_2 \in V_2$.

Then there is no path between vertices v_1 ***and*** v_2, since there is no edge joining them. This shows that G is disconnected.

Theorem 1.5 (Cont.…)

A graph *G* is disconnected if and only if its vertex set *V* can be partitioned into two nonempty, disjoint subsets V_1 *and* V_2, such that there exists no edge in *G* whose one end vertex is in V_1 and the other in V_2.

Graph G

Proof:

- Conversely, let ***G*** be a disconnected graph. Consider a vertex ***v*** in ***G***.
- Let V_1 be the set of all vertices that are joined by paths to ***v***. Since ***G*** is disconnected, V_1 does not contain all vertices of ***G***. Let V_2 be the set of the remaining vertices.
- Clearly, no vertex in V_1 is joined to any vertex in V_2 by an edge, proving the converse.

Theorem 1.6

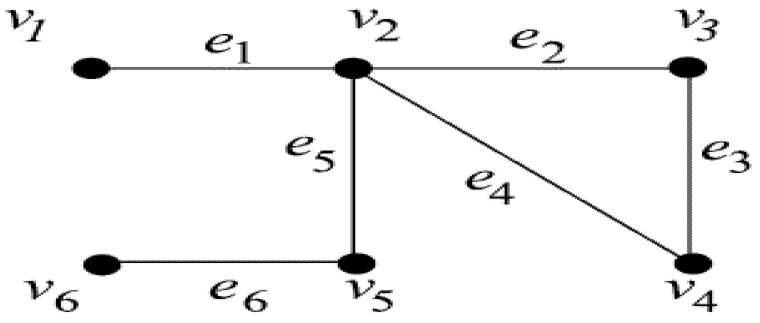
If a graph has exactly two vertices of odd degree, they must be connected by a path.

Proof:

- Let G be a graph with all its vertices of even degree, except for v_1 and v_2 which are of odd degree. Consider the component C to which v_1 belongs. Then C has an even number of vertices of odd degree. Therefore C must contain v_2, the only other vertex of odd degree.
- Thus v_1 and v_2 are in the same component, and since a component is connected, there is a path between v_1 and v_2.

Path1 - $v_1e_1v_2e_2v_3e_3v_4$

Path2 - $v_1e_1v_2e_5v_5e_6v_6$



Graph G

Theorem 1.7

A graph with n vertices and k components cannot have more than

1/2 (n−k)(n−k+1) edges.

Lemma 1.1 For any set of positive integers $n_1, n_2, \ldots, n_k$

$$\sum_{i=1}^{k} n_i^2 \leq \left(\sum_{i=1}^{k} n_i\right)^2 - (k-1)\left(2\sum_{i=1}^{k} n_i - k\right).$$

Proof The number of components is k and let the number of vertices in ith component be n_i, $1 \leq i \leq k$.

So, $\quad n_1 + n_2 + \ldots + n_k = \sum_{i=1}^{k} n_i = n.$

Now, a component with n_i vertices will have the maximum possible number of edges when it is complete, and in that case it has $\frac{1}{2}n_i(n_i - 1)$ edges.

Thus the maximum number of edges in $G = \frac{1}{2}\sum_{i=1}^{k} n_i(n_i - 1)$

$$= \frac{1}{2}\sum_{i=1}^{k} n_i^2 - \frac{1}{2}\sum_{i=1}^{k} n_i \leq \frac{1}{2}\left\{\left(\sum_{i=1}^{k} n_i\right)^2 - (k-1)\left(2\sum_{i=1}^{k} n_i - k\right)\right\} - \frac{1}{2}\sum_{i=1}^{k} n_i$$

$$= \frac{1}{2}\left\{n^2 - (k-1)(2n-k)\right\} - \frac{1}{2}n = \frac{1}{2}\left[n^2 - 2nk + k^2 + n - k\right] = \frac{1}{2}(n-k)(n-k+1).$$

Operations on Graph

Union ($G_1 \cup G_2$)

Intersection ($G_1 \cap G_2$)

Ring sum ($G_1 \oplus G_2$)

Decomposition

Deletion

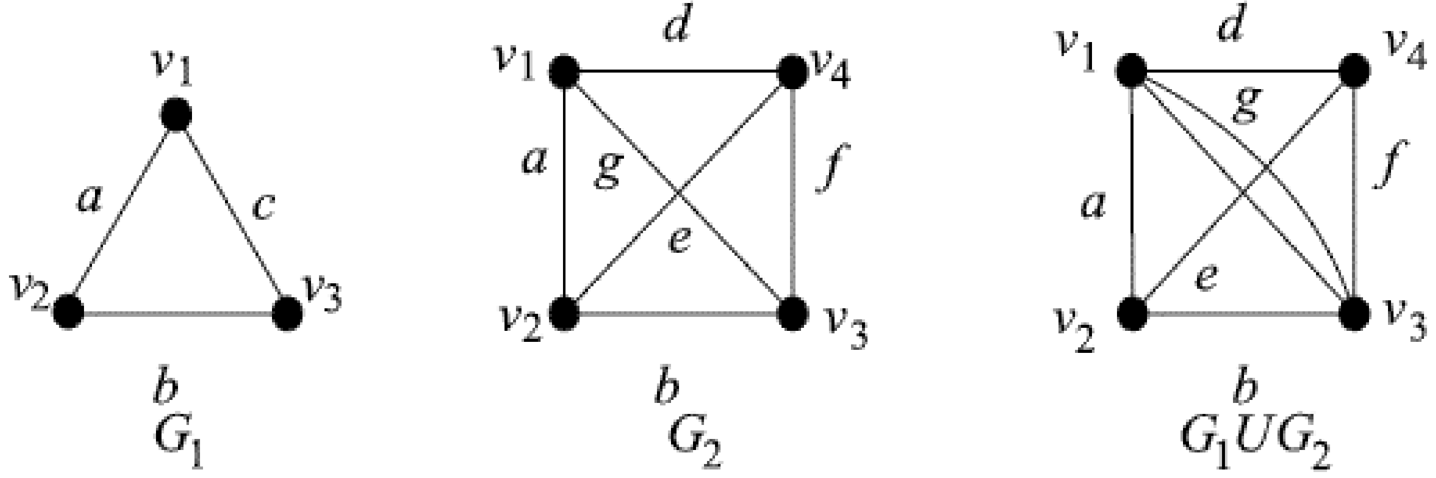
Fusion

Topological operations

Operations on Graph

Union ($G_1 \cup G_2$)

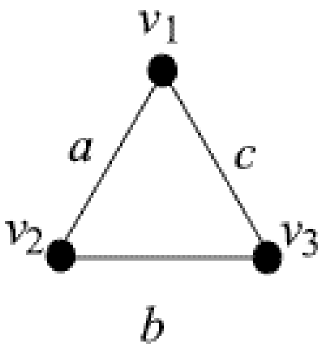
- Let $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ be two graphs.
- The union of G_1 and G_2, denoted by $G_1 \cup G_2$,is the graph whose vertex set is $V_1 \cup V_2$ and edge set is $E_1 \cup E_2$.



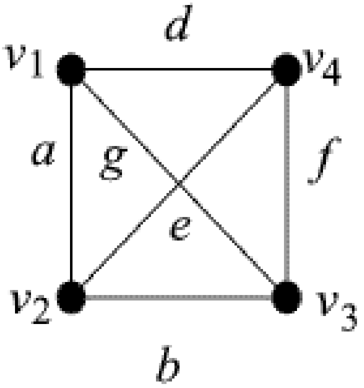
Operations on Graph

Intersection ($G_1 \cap G_2$)

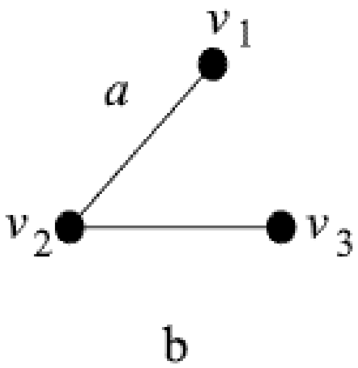
- The intersection of G_1 and G_2, denoted by $G_1 \cap G_2$, is the graph consisting only of those vertices and edges that are both in G_1 and G_2.
- Clearly, $G_1 \cap G_2 = (V_1 \cap V_2, E_1 \cap E_2)$.



G_1



G_2

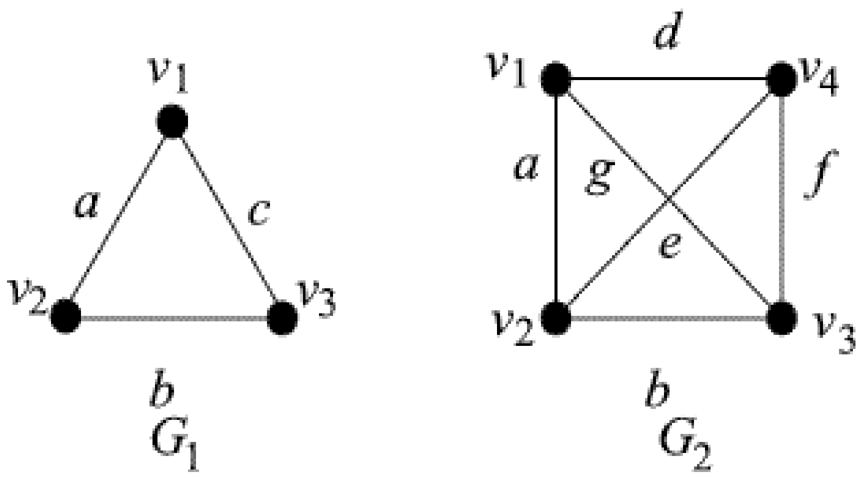


$G_1 \cap G_2$

Operations on Graph

Ring sum ($G_1 \oplus G_2$)

- The ring sum of G_1 and G_2, denoted by $G_1 \oplus G_2$, is a graph whose vertex set is $V_1 \cup V_2$ and whose edges are that of either G_1 or G_2, but not of both.



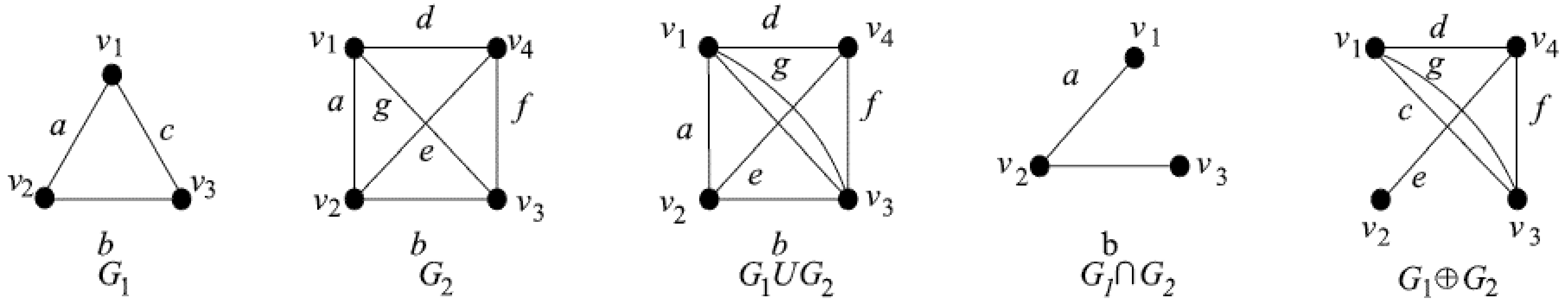
Operations on Graph

- These three operations are commutative
  1. $G_1 \cup G_2 = G_2 \cup G_1$
  2. $G_1 \cap G_2 = G_2 \cap G_1$
  3. $G_1 \oplus G_2 = G_2 \oplus G_1$
- If G_1 and G_2 are edge disjoint,

  then $G1 \cap G2$ is a null graph and $G1 \oplus G2 = G1 \cup G2$.
- If G1 and G2 are vertex disjoint,

  then $G1 \cap G2$ is empty. For any graph G,

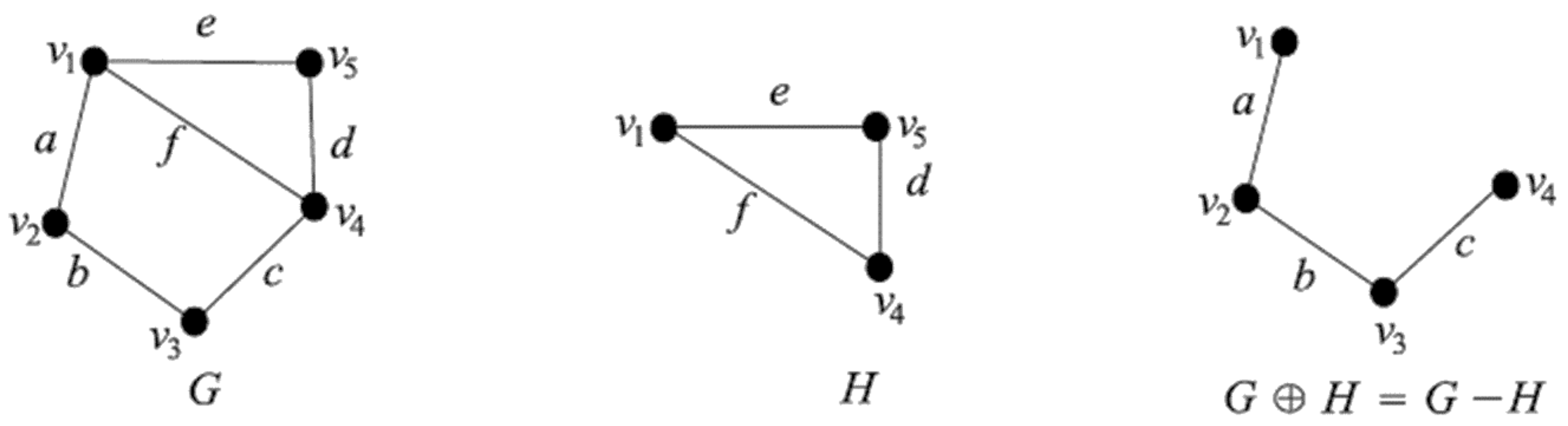
  $G \cup G = G \cap G = G$ and $G \oplus G$ = null graph.



Operations on Graph

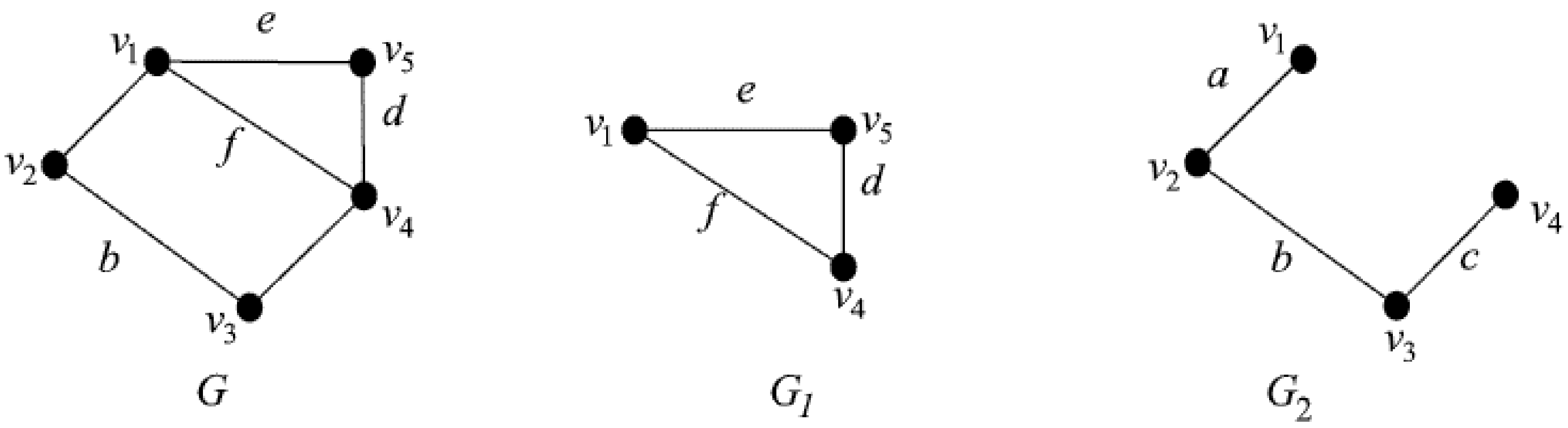
- If H is a subgraph of G, then G $\bigoplus$ H is by definition, that subgraph of G which remains after all the edges in H have been removed from G.
- We write,

$$G \bigoplus H = G - H, \text{ whenever } H \subseteq G. \; G \bigoplus H = G - H \text{ is also called complement of H in G.}$$



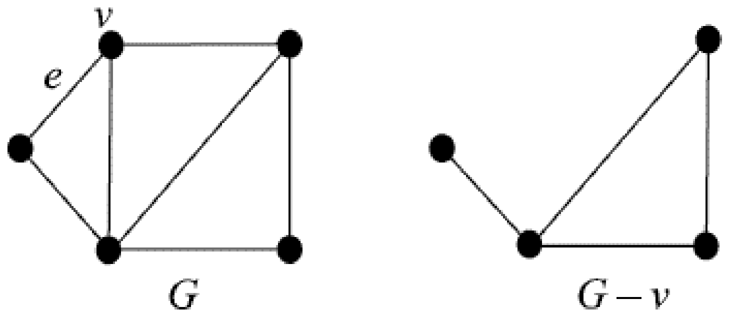
Operations on Graph - Decomposition

- A graph G is said to be decomposed into two subgraphs G_1 and G_2
  if $G_1 \cup G_2 = G$ and $G_1 \cap G_2$ = a null graph.
- In other words, every edge of G occurs either in G_1 or in G_2, but not in both, while as some of the vertices can occur in both G_1 and G_2.
- In decomposition, isolated vertices are disregarded.

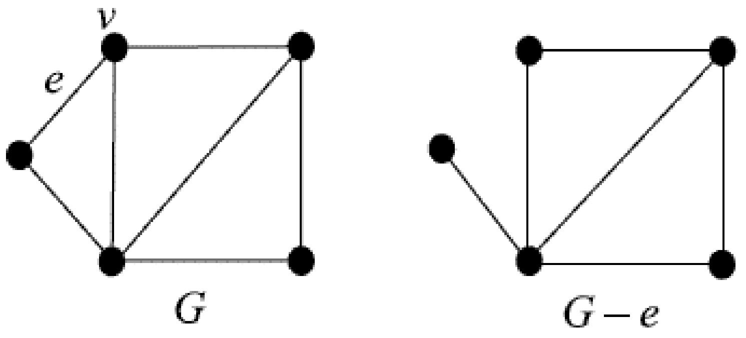


Operations on Graph - Deletion

- Let G be a graph and v be any vertex in G. Then G−v denotes the subgraph of G by deleting vertex v, and all the edges of G which are incident with v.

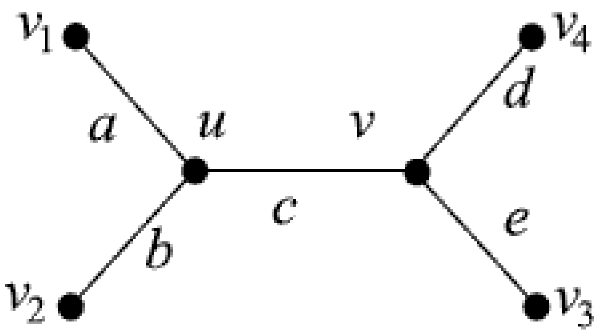


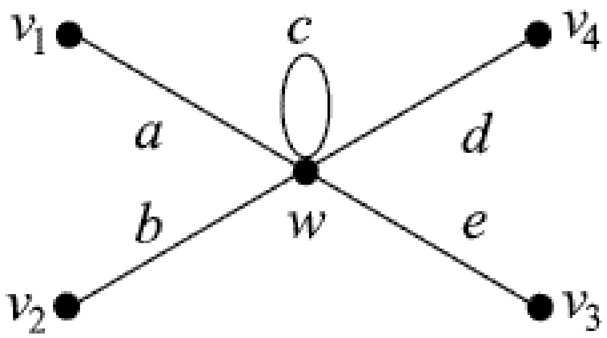
- If e is any edge of G, then G−e is a subgraph of G btained by deleting e from G.
- Deletion of an edge does not imply deletion of its end vertices. Therefore G−e = G ⊕ e .



Operations on Graph - Fusion

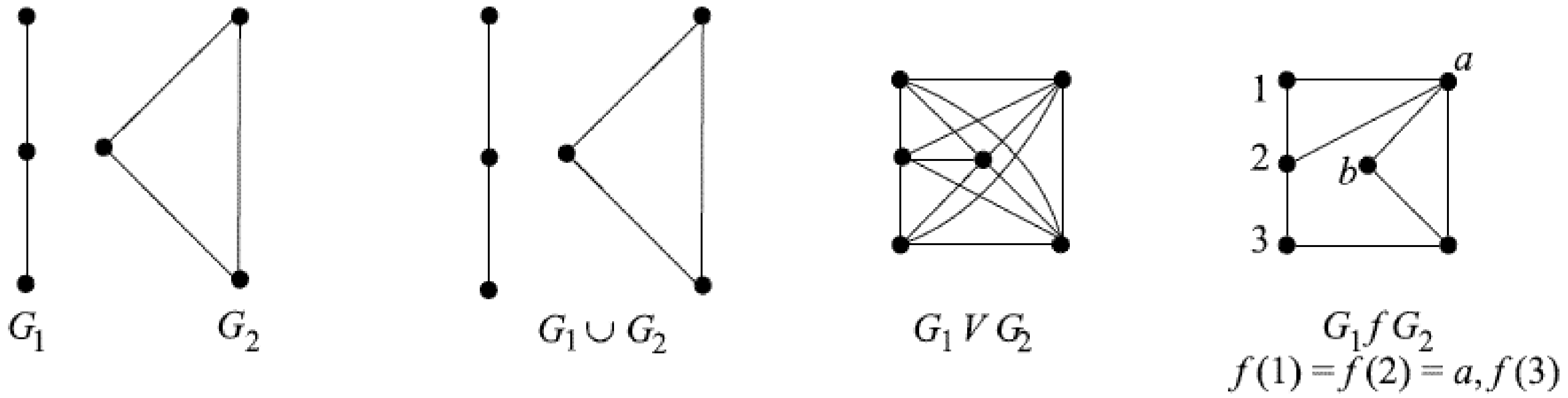
- A pair of vertices u and v in a graph are said to be fused (merged or identified) if u and v are replaced by a single new vertex such that every edge incident on u or v is incident on this new vertex.
- Therefore, fusion of vertices does not alter the number of edges, but reduces the number of vertices by one. vertices u and v are fused to a single vertex w.





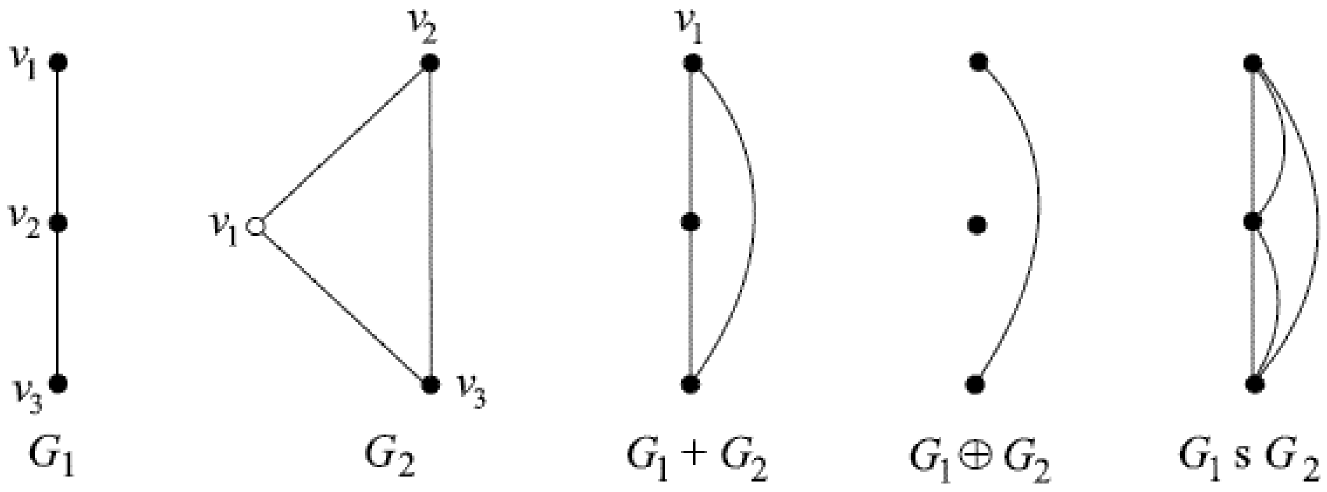
Operations on Graph: Function – Homomorphism Graph

- Let $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ be two graphs with $V_1 \cap V_2 = \Phi$ and E1 ∩ E2 = Φ.
- Union of G1 and G2, denoted by G1 U G2. If $\pi = V_1 \times V_2$ (that is all edges between V_1 and V_2), we get the join of G_1 and G_2, denoted by G_1 U G2.
- If π is a function from V_1 to V_2, we have the function graph G_1 f G_2.
- If π defines a homomorphism Φ from G_1 to G_2, we have the homomorphism graph G_1 Φ G_2. If $G_1 = G_2 = G$ (say) and p is a bijection a of V_1 to V_2, we have a permutation graph G a G.



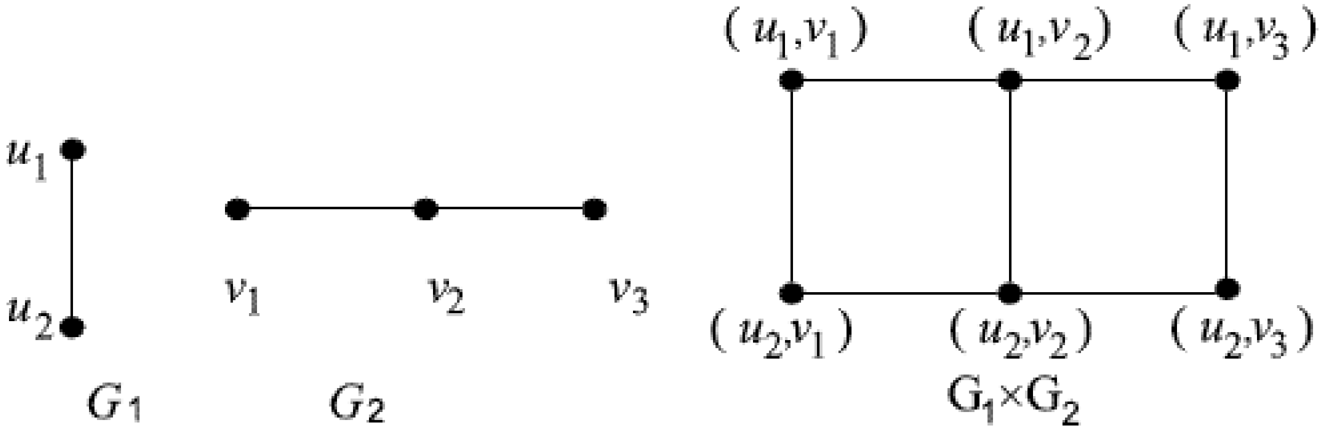
Operations on Graph: Superposition Graph

- The superposition graph $G = G_1 \text{ s } G_2$ has vertex set $V = V_1 \cup V2$ and the edge set E contains all the edges of G_1 and G_2 with the identity of edges of G_1 and G_2 in G being reserved by assigning two different labels to these edges.
- So, if v_iv_j is an edge in both G_1 and G_2, then there are two edges v_iv_j in G with different labels.



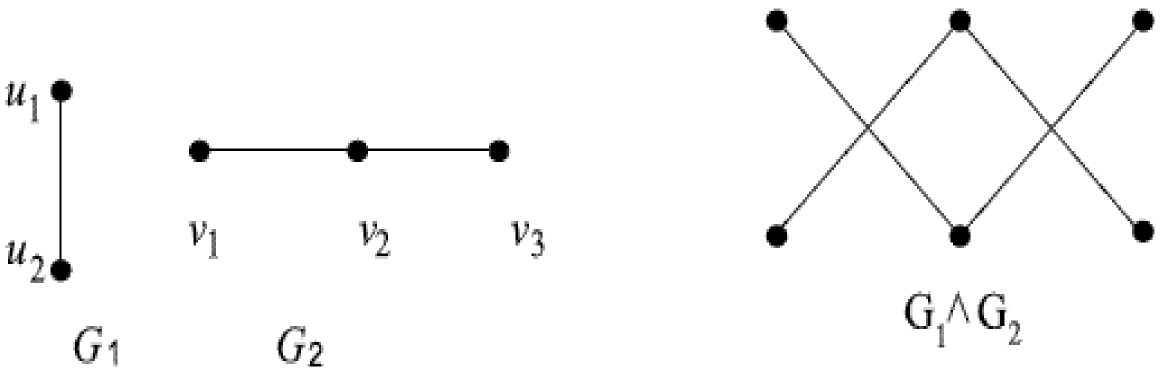
Operations on Graph: Cartesian Product

- The Cartesian product of G_1 and G_2 is denoted by $G = G_1 \times G_2$.
- It is the graph whose vertex set is $V = V_1 \times V_2$ and for any two vertices $w_1 = (u_1, v_1)$ and $w_2 = (u_2, v_2)$ in V, $u_1, u_2 \in V1$ and $v_1, v_2 \in V_2$, there is an edge $w_1w_2 \in E(G)$ if and only if either (a) $u_1 = u_2$ and $v_1v_2 \in E_2$, or (b) $v_1 = v_2$ and $u_1u_2 \in E_1$.



Operations on Graph: Tensor Product

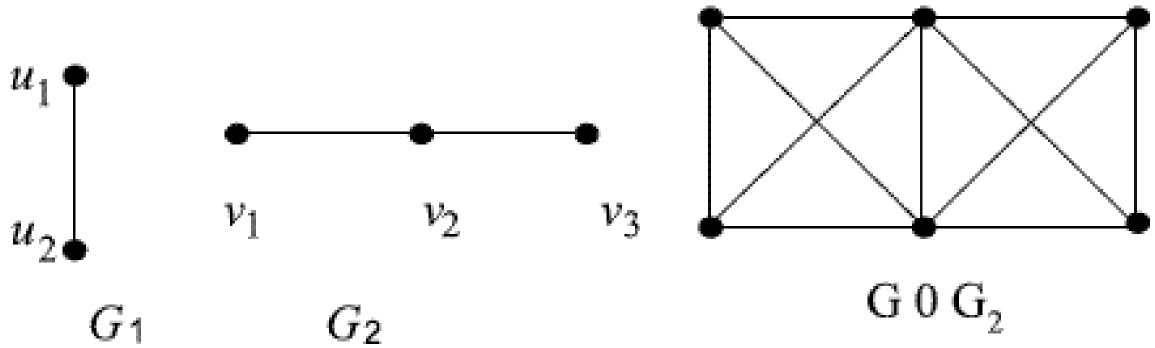
- The tensor product (conjunction) is denoted by $G = G_1 \wedge G_2$.
- It is the graph with vertex set $V = V_1 \times V_2$ and for any two vertices $w_1 = (u_1, v_1)$ and $w_2 = (u_2, v_2)$ in V; $u_1, u_2 \in V_1$ and $v_1, v_2 \in V_2$, there is an edge $w_1w_2 \in E(G)$ if and only if $u_1u_2 \in E_1$ and $v_1v_2 \in E_2$.



Operations on Graph: Normal Product

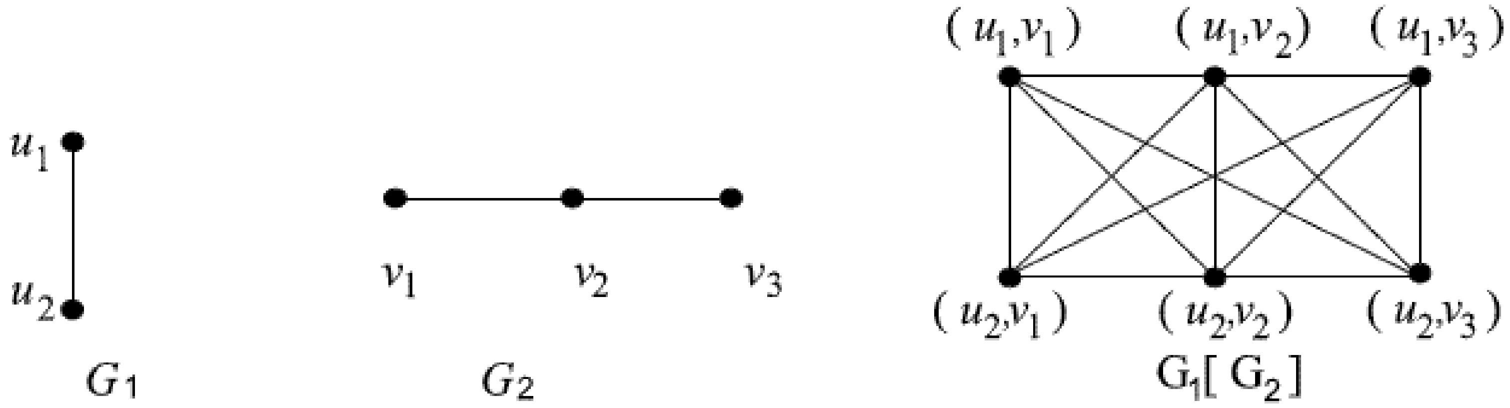
- The normal product (strong product) is denoted by $G = G_1 \circ G_2$.
- It is the graph with vertex set $V = V_1 \times V_2$ and for any two vertices $w_1 = (u_1, v_1)$ and $w_2 = (u_2, v_2)$ in V; $u_1, u_2 \in V_1$ and $v_1, v_2 \in V_2$, there is an edge $w_1w_2 \in E(G)$ if and only if one of the following holds:

(a) $u_1 = u_2$ and $v_1v_2 \in E_2$ (b) $v_1 = v_2$ and $u_1u_2 \in E_1$ (c) $u_1u_2 \in E_1$ and $v_1v_2 \in E_2$.



Operations on Graph: Composition

- The composition of G_1 and G_2 is denoted by $G = G_1[G_2]$
- It is the graph with vertex set $V = V_1 \times V_2$, and for any two vertices $w_1 = (u_1, v_1)$ and $w_2 = (u_2, v_2)$ in V; $u_1, u_2 \in V_1$ and $v_1, v_2 \in V_2$, there is an edge $w_1w_2 \in E(G)$ if and only if either (a) $u_1u_2 \in E_1$ or (b) $u_1 = u_2$ and $v_1v_2 \in E_2$.

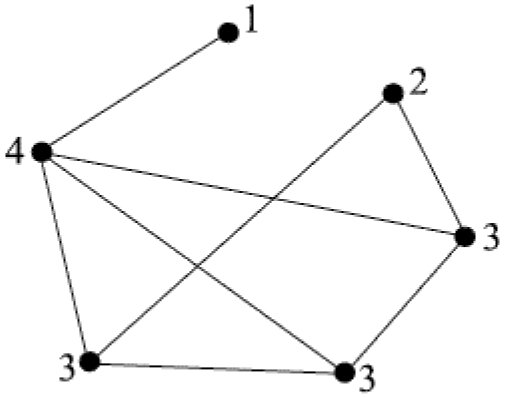


Degree Sequence

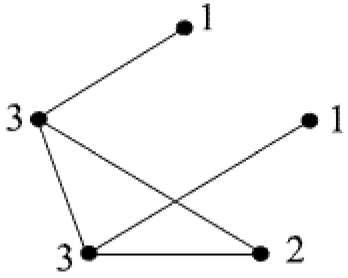
- Let d_i, $1 \le i \le n$, be the degrees of the vertices v_i of a graph in any order.
- The sequence $[d_i]^n$ 1 is called the degree sequence of the graph.
- The non-negative sequence $[d_i]_1^n$ is called the degree sequence of the graph if it is the degree sequence of some graph, and the graph is said to realise the sequence.
- The set of distinct non-negative integers occurring in a degree sequence of a graph is called its degree set.
- A set of non-negative integers is called a degree set if it is the degree set of some graph, and the graph is said to realise the degree set.

Degree Sequence

- Two graphs with the same degree sequence are said to be degree equivalent.
- The degree sequence is D = [1, 2, 3, 3, 3, 4] or D = [1 2 3^3 4] and its degree set is {1, 2, 3, 4}

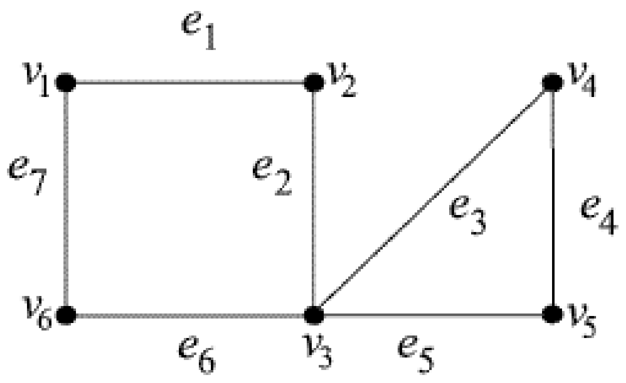


- The degree sequence of the graph is [1, 1, 2, 3, 3] and its degree set is {1, 2, 3}.

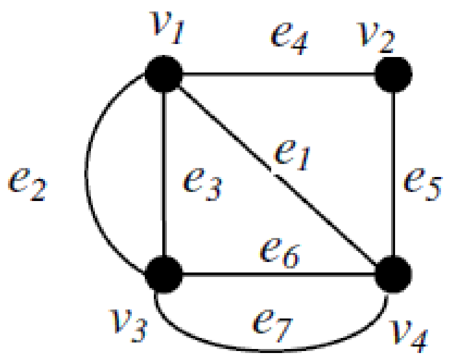


Euler Graphs

- A closed walk in a graph G containing all the edges of G is called an Euler line in G. A graph containing an Euler line is called an Euler graph.
- A path in a graph G is called Euler path if it includes every edges exactly once.
- Since the path contains every edge exactly once, it is also called Euler trail / Euler line.
- A closed Euler path is called Euler circuit. A graph which contains an Eulerian circuit is called an Eulerian graph.



1) $v_1e_1v_2e_2v_3e_3v_4e_4v_5e_5v_3e_6v_6e_7v_1$



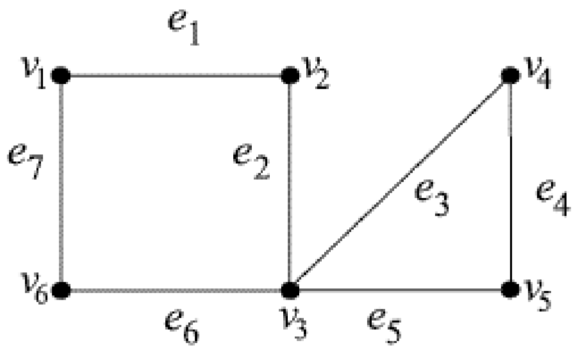
2) $v_4e_1v_1e_2v_3e_3v_1e_4v_2e_5v_4e_6v_3e_7v_4$

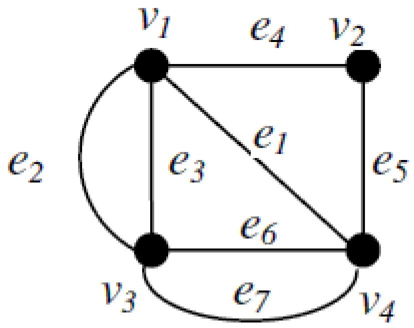
Theorem 3.1 (Euler)

A connected graph G is an Euler graph if and only if all vertices of G are of even degree.

Proof:

Suppose that *G* is an Euler graph. It therefore contains an Euler line (which is a closed walk). In tracing this walk we observe that every time the walk meets a vertex *v* it goes through two “new” edges incident on *v*—with one we “entered” *v* and with the other “exited.” This is true not only of all intermediate vertices of the walk but also of the terminal vertex, because we “exited” and “entered” the same vertex at the beginning and end of the walk, respectively. Thus if *G* is an Euler graph, the degree of every vertex is even.





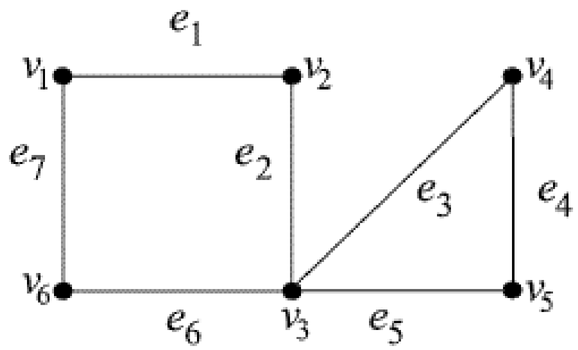
1) $v_1e_1v_2e_2v_3e_3v_4e_4v_5e_5v_3e_6v_6e_7v_1$

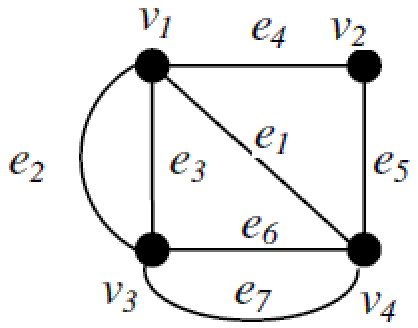
2) $v_4e_1v_1e_2v_3e_3v_1e_4v_2e_5v_4e_6v_3e_7v_4$

Theorem 3.1 (Euler)

A connected graph G is an Euler graph if and only if all vertices of G are of even degree

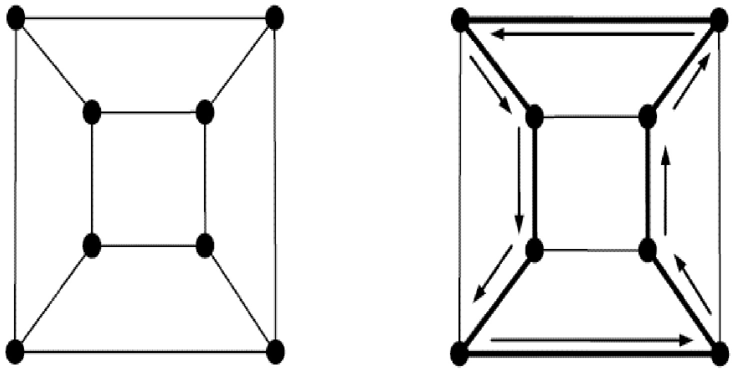
Proof: To prove the sufficiency of the condition, assume that all vertices of G are of even degree. Now we construct a walk starting at an arbitrary vertex v and going through the edges of G such that no edge is traced more than once. We continue tracing as far as possible. Since every vertex is of even degree, we can exit from every vertex we enter; the tracing cannot stop at any vertex but v. And since v is also of even degree, we shall eventually reach v when the tracing comes to an end. If this closed walk h we just traced includes all the edges of G, G is an Euler graph. If not, we remove from G all the edges in h and obtain a subgraph h' of G formed by the remaining edges. Since both G and h have all their vertices of even degree, the degrees of the vertices of h' are also even. Moreover, h' must touch h at least at one vertex a, because G is connected. Starting from a, we can again construct a new walk in graph h'. Since all the vertices of h' are of even degree, this walk in h' must terminate at vertex a; but this walk in h' can be combined with h to form a new walk, which starts and ends at vertex v and has more edges than h. This process can be repeated until we obtain a closed walk that traverses all the edges of G. Thus G is an Euler graph.

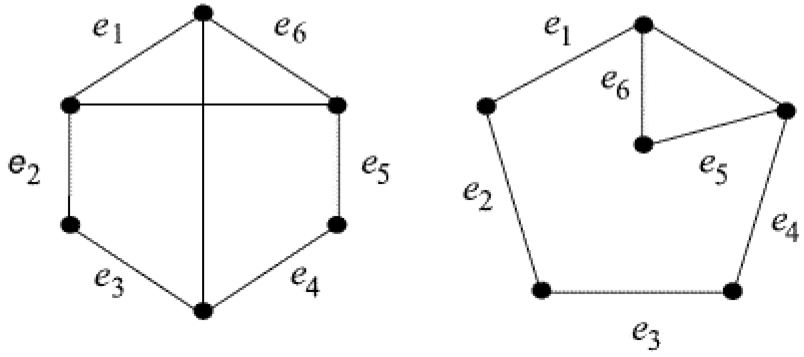




Hamiltonian Graphs

- A **Hamiltonian circuit** in a connected graph is defined as a closed walk that traverses every vertex of graph G exactly once except starting and terminal vertex.
- Removal of any one edge from a Hamiltonian circuit generates a path. This path is called **Hamiltonian path**.
- A cycle passing through all the vertices of a graph is called a Hamiltonian cycle.
- A graph containing a Hamiltonian cycle is called a Hamiltonian graph.
- A path passing through all the vertices of a graph is called a Hamiltonian path and a graph containing a Hamiltonian path is said to be traceable.

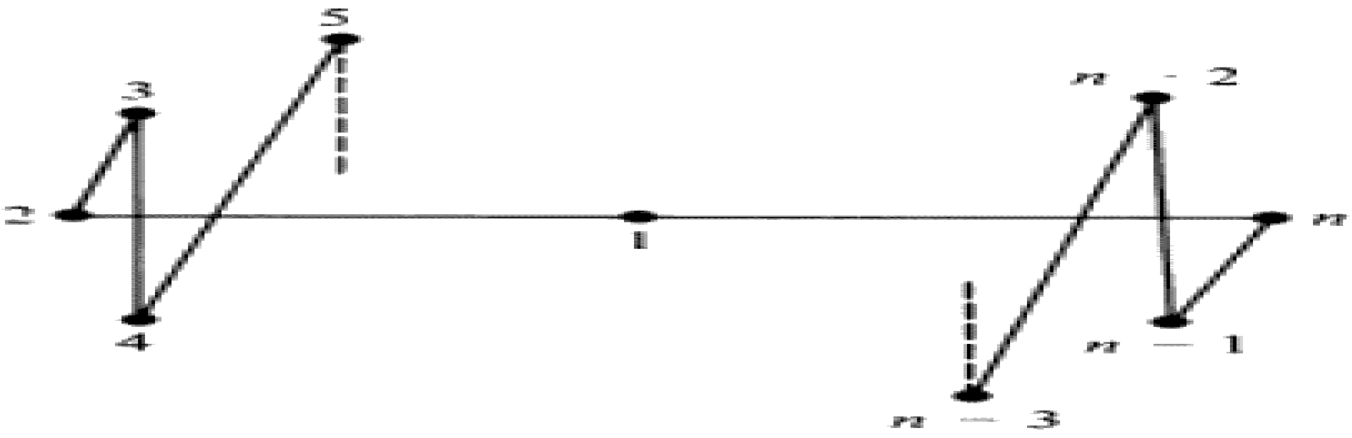




Theorem 3.2

In a complete graph with n vertices there are (n – 1) / 2 edge-disjoint Hamiltonian circuits, if n is an odd number ≥ 3.

Proof: A complete graph G of n vertices has (n – 1) / 2 edges, and a Hamiltonian circuit in G consists of n edges. Therefore, the number of edge-disjoint Hamiltonian circuits in G cannot exceed (n – 1) / 2 . That there are (n – 1) / 2 edge disjoint Hamiltonian circuits, when n is odd, can be shown as follows: The subgraph (of a complete graph of n vertices) in Figure is a Hamiltonian circuit. Keeping the vertices fixed on a circle, rotate the polygonal pattern clockwise by 360/(n – 1), 2.360/(n – 1), 3.360/(n – 1)…..(n – 3)/2.360/(n – 1) degrees. Observe that each rotation produces a Hamiltonian circuit that has no edge in common with any of the previous ones. Thus we have (n - 3)/2 new Hamiltonian circuits, all edge disjoint from the one in Figure and also edge disjoint among themselves. Hence the theorem.



REFERENCES:

1. Narsingh Deo, "Graph Theory with Applications to Engineering and Computer Science", Prentice-Hall of India Pvt. Ltd, 2003.
2. S. Pirzada, “An Introduction to Graph theory”, University Press, 2012.
3. Frank Harary, “Graph Theory”, Narosa Publishing House, 2001.
4. West D. B., “Introduction to Graph Theory”, 2nd Edition, Pearson Education, 2001.
5. Diestel R, "Graph Theory", 5th Edition, Springer, 2017.

Thank You