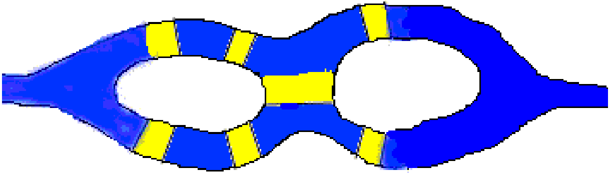
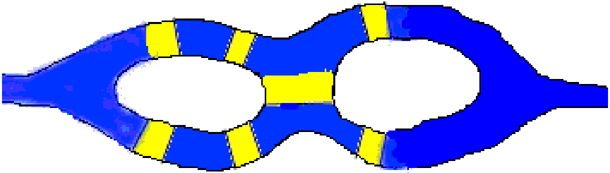
- Euler's Konigsberg Bridge Problem (18th c.)



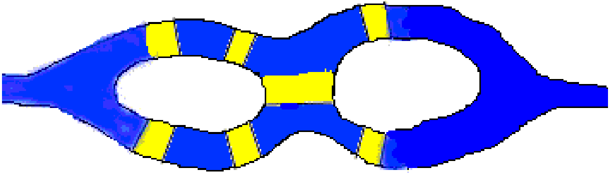
- Two islands connected to land and each other by 7 bridges:
- Can one walk through town and cross all bridges exactly once?
- Graph theory provides a way to mathematically answer that question

- Euler's Konigsberg Bridge Problem (18th c.)



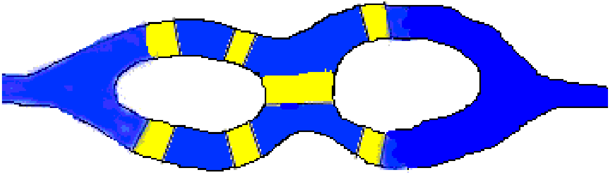
- Two islands connected to land and each other by 7 bridges:
- Can one walk through town and cross all bridges exactly once?
- Graph theory provides a way to mathematically answer that question

- Euler's Konigsberg Bridge Problem (18th c.)



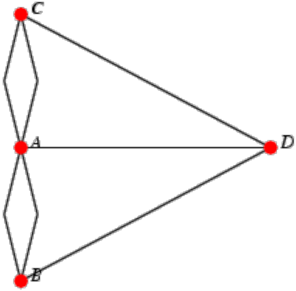
- Two islands connected to land and each other by 7 bridges:
- Can one walk through town and cross all bridges exactly once?
- Graph theory provides a way to mathematically answer that question

- Euler's Konigsberg Bridge Problem (18th c.)



- Two islands connected to land and each other by 7 bridges:
- Can one walk through town and cross all bridges exactly once?
- Graph theory provides a way to mathematically answer that question

- The Konigsberg problem can be represented by a graph



Red Dot – Vertices–islands
Lines – Edges – bridges

Definition

A graph G is an ordered triple $(V(G), E(G), \psi_G)$ consisting of

1. $V(G) \neq \emptyset$ –set of vertices
2. $E(G)$ –set of edges
3. ψ_G – An incidence function that associates with each edge of G an unordered pair of (not necessarily distinct) vertices of G.

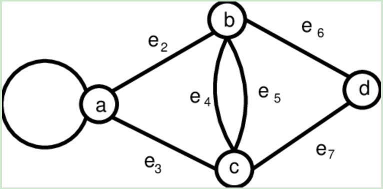
If e is an edge and u and v are vertices such that

$$\psi_G(e) = uv$$

then e is said to join u and v, the vertices u and v are called the ends of e.

Example

$$V(G) = \{a, b, c, d\},\ E(G) = \{e_i\ :\ 1 \leq i \leq 7\}$$

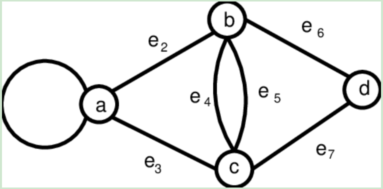


$$\psi_G(e_1) = aa,\ \psi_G(e_2) = ab,\ \psi_G(e_3) = ac,\ \psi_G(e_4) = bc,$$

$$\psi_G(e_5) = bc,\ \psi_G(e_6) = bd,\ \psi_G(e_7) = cd$$

Example

$$V(G) = \{a, b, c, d\},\ E(G) = \{e_i\ :\ 1 \leq i \leq 7\}$$



$$\psi_G(e_1) = aa,\ \psi_G(e_2) = ab,\ \psi_G(e_3) = ac,\ \psi_G(e_4) = bc,$$
$$\psi_G(e_5) = bc,\ \psi_G(e_6) = bd,\ \psi_G(e_7) = cd$$

Some Terminologies

- The ends of an edge are said to be incident with the edge, and vice versa.

- Two vertices which are incident with a common edge are adjacent.

- Two edges which are incident with a common vertex are also called adjacent.

- An edge with identical ends is called a loop.

- An edge with distinct ends a link.

- If more than one link share the same pair of vertices, then they are called as multiple edges.

Some Terminologies

- The ends of an edge are said to be incident with the edge, and vice versa.
- Two vertices which are incident with a common edge are adjacent.
- Two edges which are incident with a common vertex are also called adjacent.
- An edge with identical ends is called a loop.
- An edge with distinct ends a link.
- If more than one link share the same pair of vertices, then they are called as multiple edges.

- The ends of an edge are said to be incident with the edge, and vice versa.
- Two vertices which are incident with a common edge are adjacent.
- Two edges which are incident with a common vertex are also called adjacent.
- An edge with identical ends is called a loop.
- An edge with distinct ends a link.
- If more than one link share the same pair of vertices, then they are called as multiple edges.

- The ends of an edge are said to be incident with the edge, and vice versa.
- Two vertices which are incident with a common edge are adjacent.
- Two edges which are incident with a common vertex are also called adjacent.
- An edge with identical ends is called a loop.
- An edge with distinct ends a link.
- If more than one link share the same pair of vertices, then they are called as multiple edges.

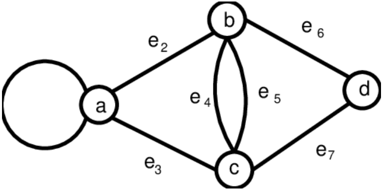
Some Terminologies

- The ends of an edge are said to be incident with the edge, and vice versa.
- Two vertices which are incident with a common edge are adjacent.
- Two edges which are incident with a common vertex are also called adjacent.
- An edge with identical ends is called a loop.
- An edge with distinct ends a link.
- If more than one link share the same pair of vertices, then they are called as multiple edges.

Some Terminologies

- The ends of an edge are said to be incident with the edge, and vice versa.
- Two vertices which are incident with a common edge are adjacent.
- Two edges which are incident with a common vertex are also called adjacent.
- An edge with identical ends is called a loop.
- An edge with distinct ends a link.
- If more than one link share the same pair of vertices, then they are called as multiple edges.

Example



- Vertex b is incident with edges e_2, e_4, e_5, e_6.
- Vertices a and b are adjacent.
- Edges e_2 and e_4 are adjacent.
- e_1 is a loop and all other edges are links.

- A graph is finite if both its vertex set and edge set are finite. In this course we study only finite graphs.
- A graph with just one vertex is called trivial and all other graphs are nontrivial.
- A graph is simple if it has no loops and no multiple edges.
- $\nu(G) = |V(G)|$ and $\epsilon(G) = |E(G)|$

- A graph is finite if both its vertex set and edge set are finite. In this course we study only finite graphs.
- A graph with just one vertex is called trivial and all other graphs are nontrivial.
- A graph is simple if it has no loops and no multiple edges.
- $\nu(G) = |V(G)|$ and $\epsilon(G) = |E(G)|$

Special Graphs

- A graph is finite if both its vertex set and edge set are finite. In this course we study only finite graphs.
- A graph with just one vertex is called trivial and all other graphs are nontrivial.
- A graph is simple if it has no loops and no multiple edges.
- $\nu(G) = |V(G)|$ and $\epsilon(G) = |E(G)|$

Special Graphs

- A graph is finite if both its vertex set and edge set are finite. In this course we study only finite graphs.
- A graph with just one vertex is called trivial and all other graphs are nontrivial.
- A graph is simple if it has no loops and no multiple edges.
- $\nu(G) = |V(G)|$ and $\epsilon(G) = |E(G)|$

- Degree $d_G(v)$ of a vertex v in G is the number of edges of G incident with v.
- Each loop counting two edges towards the degree.
- $\delta(G)$= minimum degree.
- $\Delta(G)$ = maximum degree.

Theorem

$$\sum_{v \in V(G)} d_G(v) = 2|E|$$

Proof : Exercise.

Vertex Degrees

- Degree $d_G(v)$ of a vertex v in G is the number of edges of G incident with v.
- Each loop counting two edges towards the degree.
- $\delta(G)$= minimum degree.
- $\Delta(G)$ = maximum degree.

Theorem

$$\sum_{v \in V(G)} d_G(v) = 2|E|$$

Proof : Exercise.

- Degree $d_G(v)$ of a vertex v in G is the number of edges of G incident with v.
- Each loop counting two edges towards the degree.
- $\delta(G)$= minimum degree.
- $\Delta(G)$ = maximum degree.

Theorem

$$\sum_{v \in V(G)} d_G(v) = 2|E|$$

Proof : Exercise.

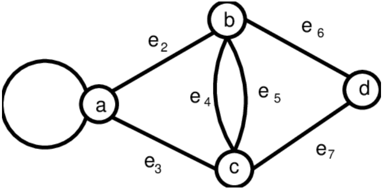
- Degree $d_G(v)$ of a vertex v in G is the number of edges of G incident with v.
- Each loop counting two edges towards the degree.
- $\delta(G)$= minimum degree.
- $\Delta(G)$ = maximum degree.

Theorem

$$\sum_{v \in V(G)} d_G(v) = 2|E|$$

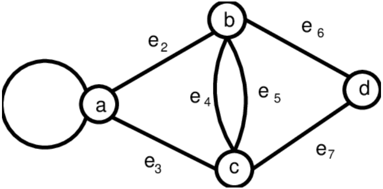
Proof : Exercise.

Example



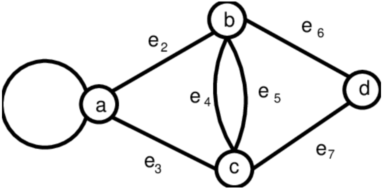
- Degree of $a : 4,\ \ b : 4,\ \ c\ :\ 4,\ \ d\ :\ 2$
- Maximum degree : $\Delta(G) = 4$
- Minimum degree : $\delta(G) = 2$

Example



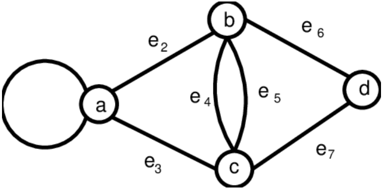
- Degree of $a : 4$, $b : 4$, $c : 4$, $d : 2$
- Maximum degree : $\Delta(G) = 4$
- Minimum degree : $\delta(G) = 2$

Example



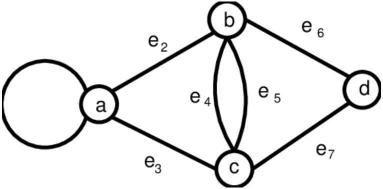
- Degree of $a : 4,\ \ b : 4,\ \ c\ :\ 4,\ \ d\ :\ 2$
- Maximum degree : $\Delta(G) = 4$
- Minimum degree : $\delta(G) = 2$

Example

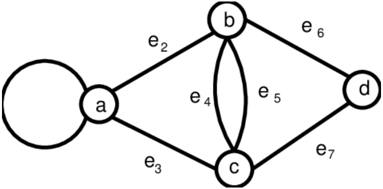


- Degree of $a : 4$, $b : 4$, $c : 4$, $d : 2$
- Maximum degree : $\Delta(G) = 4$
- Minimum degree : $\delta(G) = 2$

Example

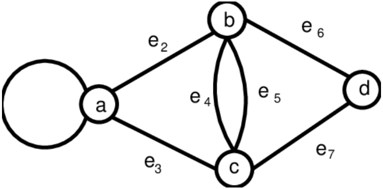


- Degree of $a : 4,\ \ b : 4,\ \ c\ :\ 4,\ \ d\ :\ 2$
- Maximum degree : $\Delta(G) = 4$
- Minimum degree : $\delta(G) = 2$



- Degree of a : 4, b : 4, c : 4, d : 2
- Maximum degree : $\Delta(G) = 4$
- Minimum degree : $\delta(G) = 2$

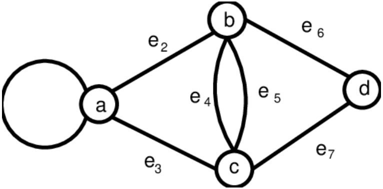
Example



- Degree of $a : 4,\ \ b : 4,\ \ c\ :\ 4,\ \ d\ :\ 2$
- Maximum degree : $\Delta(G) = 4$
- Minimum degree : $\delta(G) = 2$

Definition

G has vertices $v_1, v_2, \cdots, v_n$ the sequence $(d(v_1), d(v_2), \cdots, d(v_n))$ is called a degree sequence of G where $d(v_i) \leq d(v_{i+1})$.



Degree sequence of the above graph : $(4, 4, 4, 2)$.

- Does there exist a graph with the following sequence?

$$(4, 4, 4, 3)??$$

- Ans : No.

Theorem

A sequence $(d_1, d_2, \cdots, d_n)$ is graphical (Multigraphs) iff $\sum_i d_i$ is even.

Proof.

Hint : Induction on n. □

- Does there exist a graph with the following sequence?

$$(4,4,4,3)??$$

- Ans : No.

Theorem

A sequence $(d_1, d_2, \cdots, d_n)$ is graphical (Multigraphs) iff $\sum_i d_i$ is even.

Proof.

Hint : Induction on n. □

- Does there exist a graph with the following sequence?

$$(4, 4, 4, 3)??$$

- Ans : No.

Theorem

A sequence $(d_1, d_2, \cdots, d_n)$ is graphical (Multigraphs) iff $\sum_i d_i$ is even.

Proof.

Hint : Induction on n. □

- Does there exist a graph with the following sequence?

$$(4, 4, 4, 3)??$$

- Ans : No.

Theorem

A sequence $(d_1, d_2, \cdots, d_n)$ is graphical (Multigraphs) iff $\sum_i d_i$ is even.

Proof.

Hint : Induction on n. □

Degree sequence-simple graph

- Does there exist a simple graph with the following sequence?

$$(4, 3, 2, 1)??$$

- Ans : No.

Theorem

Havel-Hakimi

A sequence $(s, t_1, t_2, \cdots, t_s, d_1, d_2, \cdots, d_k)$ is graphical (simple) iff $(t_1 - 1, t_2 - 1, \cdots, t_s - 1, d_1, d_2, \cdots, d_k)$ is graphical.

Proof.

Hint : Edge swap □

- Does there exist a simple graph with the following sequence?

$$(4, 3, 2, 1)??$$

- Ans : No.

Theorem

Havel-Hakimi

A sequence $(s, t_1, t_2, \cdots, t_s, d_1, d_2, \cdots, d_k)$ is graphical (simple) iff $(t_1 - 1, t_2 - 1, \cdots, t_s - 1, d_1, d_2, \cdots, d_k)$ is graphical.

Proof.

Hint : Edge swap □

- Does there exist a simple graph with the following sequence?

$$(4, 3, 2, 1)??$$

- Ans : No.

Theorem

Havel-Hakimi

A sequence $(s, t_1, t_2, \cdots, t_s, d_1, d_2, \cdots, d_k)$ is graphical (simple) iff $(t_1 - 1, t_2 - 1, \cdots, t_s - 1, d_1, d_2, \cdots, d_k)$ is graphical.

Proof.

Hint : Edge swap □

Degree sequence-simple graph

- Does there exist a simple graph with the following sequence?

$$(4, 3, 2, 1)??$$

- Ans : No.

Theorem

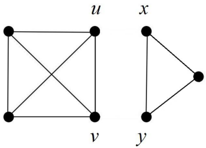
Havel-Hakimi

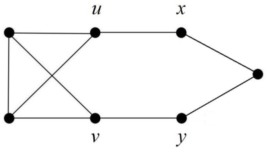
A sequence $(s, t_1, t_2, \cdots, t_s, d_1, d_2, \cdots, d_k)$ is graphical (simple) iff $(t_1 - 1, t_2 - 1, \cdots, t_s - 1, d_1, d_2, \cdots, d_k)$ is graphical.

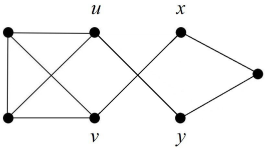
Proof.

Hint : Edge swap □

Edge-Swap







Example

Is the following sequence Graphical :

$(5, 5, 5, 3, 3, 3, 3, 3)$

$\Leftrightarrow (\star, 5-1, 5-1, 3-1, 3-1, 3-1, 3, 3) = (4, 4, 3, 3, 2, 2, 2)$

$\Leftrightarrow (\star, 4-1, 3-1, 3-1, 2-1, 2, 2) = (3, 2, 2, 2, 2, 1)$

$\Leftrightarrow (\star, 2-1, 2-1, 2-1, 2, 1) = (2, 1, 1, 1, 1)$

$\Leftrightarrow (\star, 1-1, 1-1, 1, 1) = (1, 1, 0, 0)$

- $(1, 1, 0, 0)$ is a graph with 4 vertices and one edge.

Example

Is the following sequence Graphical :

$(3, 3, 3, 2)$??

$\Leftrightarrow (\star, 3-1, 3-1, 2-1) = (2, 2, 1)$

$\Leftrightarrow (\star, 1, 0) = (1, 0)$

- $(1, 0)$ is not graphical.

Subgraphs

- A graph H is a subgraph of G (written $H \subset G$) if
  1. $V(H) \subset V(G)$,
  2. $E(H) \subset E(G)$
  3. ψ_H is the restriction of ψ_G to $E(H)$.

- Proper Subgraph : When $H \subseteq G$ but $H \neq G$, we write $H \subset G$ and call H a proper subgraph of G.

- Supergraph : If H is a subgraph of G, then G is called a supergraph of H.

- A spanning subgraph (or spanning supergraph) of G is a subgraph (or supergraph) H with $V(H) = V(G)$.

- A graph H is a subgraph of G (written $H \subset G$) if
  1. $V(H) \subset V(G)$,
  2. $E(H) \subset E(G)$
  3. ψ_H is the restriction of ψ_G to $E(H)$.

- Proper Subgraph : When $H \subseteq G$ but $H \neq G$, we write $H \subset G$ and call H a proper subgraph of G.

- Supergraph : If H is a subgraph of G, then G is called a supergraph of H.

- A spanning subgraph (or spanning supergraph) of G is a subgraph (or supergraph) H with $V(H) = V(G)$.

- A graph H is a subgraph of G (written $H \subset G$) if
  1. $V(H) \subset V(G)$,
  2. $E(H) \subset E(G)$
  3. ψ_H is the restriction of ψ_G to $E(H)$.

- Proper Subgraph : When $H \subseteq G$ but $H \neq G$, we write $H \subset G$ and call H a proper subgraph of G.

- Supergraph : If H is a subgraph of G, then G is called a supergraph of H.

- A spanning subgraph (or spanning supergraph) of G is a subgraph (or supergraph) H with $V(H) = V(G)$.

Subgraphs

- A graph H is a subgraph of G (written $H \subset G$) if
  1. $V(H) \subset V(G)$,
  2. $E(H) \subset E(G)$
  3. ψ_H is the restriction of ψ_G to $E(H)$.

- Proper Subgraph : When $H \subseteq G$ but $H \neq G$, we write $H \subset G$ and call H a proper subgraph of G.

- Supergraph : If H is a subgraph of G, then G is called a supergraph of H.

- A spanning subgraph (or spanning supergraph) of G is a subgraph (or supergraph) H with $V(H) = V(G)$.

- A graph H is a subgraph of G (written $H \subset G$) if
  1. $V(H) \subset V(G)$,
  2. $E(H) \subset E(G)$
  3. ψ_H is the restriction of ψ_G to $E(H)$.

- Proper Subgraph : When $H \subseteq G$ but $H \neq G$, we write $H \subset G$ and call H a proper subgraph of G.

- Supergraph : If H is a subgraph of G, then G is called a supergraph of H.

- A spanning subgraph (or spanning supergraph) of G is a subgraph (or supergraph) H with $V(H) = V(G)$.

Subgraphs

- A graph H is a subgraph of G (written $H \subset G$) if
  1. $V(H) \subset V(G)$,
  2. $E(H) \subset E(G)$
  3. ψ_H is the restriction of ψ_G to $E(H)$.

- Proper Subgraph : When $H \subseteq G$ but $H \neq G$, we write $H \subset G$ and call H a proper subgraph of G.

- Supergraph : If H is a subgraph of G, then G is called a supergraph of H.

- A spanning subgraph (or spanning supergraph) of G is a subgraph (or supergraph) H with $V(H) = V(G)$.

Subgraphs

- A graph H is a subgraph of G (written $H \subset G$) if
  1. $V(H) \subset V(G)$,
  2. $E(H) \subset E(G)$
  3. ψ_H is the restriction of ψ_G to $E(H)$.

- Proper Subgraph : When $H \subseteq G$ but $H \neq G$, we write $H \subset G$ and call H a proper subgraph of G.

- Supergraph : If H is a subgraph of G, then G is called a supergraph of H.

- A spanning subgraph (or spanning supergraph) of G is a subgraph (or supergraph) H with $V(H) = V(G)$.

Subgraphs (Induced)

▶ Induced subgraph : $G' = G[V'] \ : \ V' \subseteq V$,

$E(G')$ is all edges of G that have both ends in V'.

▶ Induced subgraph : $G[V \setminus V']$: Denoted by $G \setminus V'$

subgraph obtained from G by deleting the vertices in V' together with their incident edges.

If $V' = \{v\}$ a singleton we simply write $G - v$.

▶ Induced subgraph : $G' = G[V']$: $V' \subseteq V$,

$E(G')$ is all edges of G that have both ends in V'.

▶ Induced subgraph : $G[V \setminus V']$: Denoted by $G \setminus V'$

subgraph obtained from G by deleting the vertices in V' together with their incident edges.

If $V' = \{v\}$ a singleton we simply write $G - v$.

▶ Induced subgraph : $G' = G[V']$: $V' \subseteq V$,

$E(G')$ is all edges of G that have both ends in V'.

▶ Induced subgraph : $G[V \setminus V']$: Denoted by $G \setminus V'$

subgraph obtained from G by deleting the vertices in V' together with their incident edges.

If $V' = \{v\}$ a singleton we simply write $G - v$.

▶ Edge-Induced subgraph : $G[E']$: $E' \subset E$.

$G[E']$ is the graph obtained by simply taking all edges in E' and vertices incident with them.

▶ Edge-Induced subgraph : $G \setminus E'$– subgraph obtained by deleting edges in E'. (not vertices).

▶ $G \setminus \{e\} = G - e$ and $G + \{e\} = G + e$

Edge-Induced Subgraph

► Edge-Induced subgraph : $G[E']$: $E' \subset E$.

$G[E']$ is the graph obtained by simply taking all edges in E' and vertices incident with them.

► Edge-Induced subgraph : $G \setminus E'$– subgraph obtained by deleting edges in E'. (not vertices).

► $G \setminus \{e\} = G - e$ and $G + \{e\} = G + e$

Let G_1 and G_2 be subgraphs of G.

- G_1 and G_2 are disjoint if $V(G_1) \cap V(G_2) = \emptyset$.
- G_1 and G_2 are Edge-disjoint if $E(G_1) \cap E(G_2) = \emptyset$.
- The union $H = G_1 \cup G_2 \subseteq G$ of G_1 and G_2 is the subgraph with $V(H) = V(G_1) \cup V(G_2)$ and $E(H) = E(G_1) \cup E(G_2)$.
- The intersection $H = G_1 \cap G_2 \subseteq G$ of G_1 and G_2 is the subgraph with $V(H) = V(G_1) \cap V(G_2)$ and $E(H) = E(G_1) \cap E(G_2)$.

Let G_1 and G_2 be subgraphs of G.

- G_1 and G_2 are disjoint if $V(G_1) \cap V(G_2) = \emptyset$.
- G_1 and G_2 are Edge-disjoint if $E(G_1) \cap E(G_2) = \emptyset$.
- The union $H = G_1 \cup G_2 \subseteq G$ of G_1 and G_2 is the subgraph with $V(H) = V(G_1) \cup V(G_2)$ and $E(H) = E(G_1) \cup E(G_2)$.
- The intersection $H = G_1 \cap G_2 \subseteq G$ of G_1 and G_2 is the subgraph with $V(H) = V(G_1) \cap V(G_2)$ and $E(H) = E(G_1) \cap E(G_2)$.

Let G_1 and G_2 be subgraphs of G.

- G_1 and G_2 are disjoint if $V(G_1) \cap V(G_2) = \emptyset$.
- G_1 and G_2 are Edge-disjoint if $E(G_1) \cap E(G_2) = \emptyset$.
- The union $H = G_1 \cup G_2 \subseteq G$ of G_1 and G_2 is the subgraph with $V(H) = V(G_1) \cup V(G_2)$ and $E(H) = E(G_1) \cup E(G_2)$.
- The intersection $H = G_1 \cap G_2 \subseteq G$ of G_1 and G_2 is the subgraph with $V(H) = V(G_1) \cap V(G_2)$ and $E(H) = E(G_1) \cap E(G_2)$.

Let G_1 and G_2 be subgraphs of G.

- G_1 and G_2 are disjoint if $V(G_1) \cap V(G_2) = \emptyset$.
- G_1 and G_2 are Edge-disjoint if $E(G_1) \cap E(G_2) = \emptyset$.
- The union $H = G_1 \cup G_2 \subseteq G$ of G_1 and G_2 is the subgraph with $V(H) = V(G_1) \cup V(G_2)$ and $E(H) = E(G_1) \cup E(G_2)$.
- The intersection $H = G_1 \cap G_2 \subseteq G$ of G_1 and G_2 is the subgraph with $V(H) = V(G_1) \cap V(G_2)$ and $E(H) = E(G_1) \cap E(G_2)$.

Let G_1 and G_2 be subgraphs of G.

- G_1 and G_2 are disjoint if $V(G_1) \cap V(G_2) = \emptyset$.
- G_1 and G_2 are Edge-disjoint if $E(G_1) \cap E(G_2) = \emptyset$.
- The union $H = G_1 \cup G_2 \subseteq G$ of G_1 and G_2 is the subgraph with $V(H) = V(G_1) \cup V(G_2)$ and $E(H) = E(G_1) \cup E(G_2)$.
- The intersection $H = G_1 \cap G_2 \subseteq G$ of G_1 and G_2 is the subgraph with $V(H) = V(G_1) \cap V(G_2)$ and $E(H) = E(G_1) \cap E(G_2)$.

Incidence and Adjacency Matrices

- $V(G) = \{v_1, v_2, \cdots, v_n\}$
- $E(G) = \{e_1, e_2, \cdots, e_m\}$

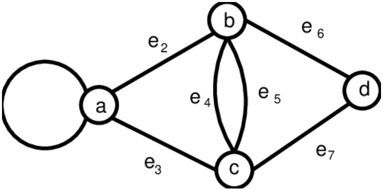
- **Incidence matrix :** $M(G) = [m_{ij}]_{n \times m}$

1. **Size :** $n \times m = |V| \times |E|$.
2. $m_{ij} =$ **number of times vertex v_i is incident with edge e_j.**
3. $0 \leq m_{ij} \leq 2$

- Adjacency matrix : $A(G) = [a_{ij}]_{n \times n}$

1. Size : $n \times n = |V| \times |V|$.
2. $a_{ij} =$ number of edges between vertex v_i vertex v_j.
3. $m_{ij} \geq 0$.

Example

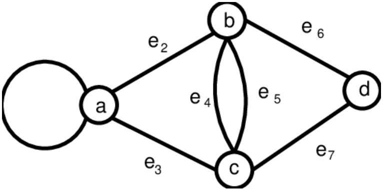


- Incidence matrix is of order 4×7 and the adjacency matrix is of order 4×4.

$$
\begin{array}{cc}
 & \begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}
\end{array}
$$

Note : Loops counted twice.

Example

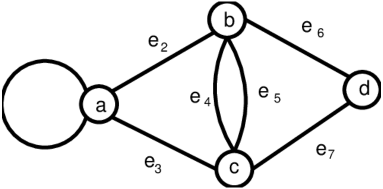


► Incidence matrix is of order 4 × 7 and the adjacency matrix is of order 4 × 4.

$$\begin{array}{c c} & \begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Note : Loops counted twice.

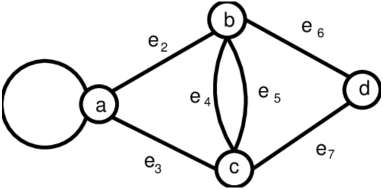
Example



- The adjacency matrix A is of order 4×4.

$$\begin{array}{c c} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Note : Loops counted only once.



- The adjacency matrix A is of order 4×4.

$$\begin{array}{cc} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Note : Loops counted only once.

- A graph G is k -regular if $d(v) = k$ for all $v \in V(G)$; a regular graph is one that is k-regular for some k.
- A simple graph is said to complete-K_n if every pair of vertices are incident with a unique edge.
- A graph is said to be Bipartite is $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$ and every edge has one in V_1 and its other end in V_2.
- A complete bipartite graph-$K_{m,n}$ is a simple bipartite graph with bipartition (X, Y) in which each vertex of X is joined to each vertex of Y; if $|X| = m$ and $|Y| = n$, such a graph is denoted by $K_{m,n}$.

Special Graphs

- A graph G is k -regular if $d(v) = k$ for all $v \in V(G)$; a regular graph is one that is k-regular for some k.
- A simple graph is said to complete-K_n if every pair of vertices are incident with a unique edge.
- A graph is said to be Bipartite is $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$ and every edge has one in V_1 and its other end in V_2.
- A complete bipartite graph-$K_{m,n}$ is a simple bipartite graph with bipartition (X, Y) in which each vertex of X is joined to each vertex of Y; if $|X| = m$ and $|Y| = n$, such a graph is denoted by $K_{m,n}$.

Special Graphs

- A graph G is k -regular if $d(v) = k$ for all $v \in V(G)$; a regular graph is one that is k-regular for some k.
- A simple graph is said to complete-K_n if every pair of vertices are incident with a unique edge.
- A graph is said to be Bipartite is $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$ and every edge has one in V_1 and its other end in V_2.
- A complete bipartite graph-$K_{m,n}$ is a simple bipartite graph with bipartition (X, Y) in which each vertex of X is joined to each vertex of Y; if $|X| = m$ and $|Y| = n$, such a graph is denoted by $K_{m,n}$.

- A graph G is k -regular if $d(v) = k$ for all $v \in V(G)$; a regular graph is one that is k-regular for some k.
- A simple graph is said to complete-K_n if every pair of vertices are incident with a unique edge.
- A graph is said to be Bipartite is $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$ and every edge has one in V_1 and its other end in V_2.
- A complete bipartite graph-$K_{m,n}$ is a simple bipartite graph with bipartition (X, Y) in which each vertex of X is joined to each vertex of Y; if $|X| = m$ and $|Y| = n$, such a graph is denoted by $K_{m,n}$.

Graph Isomorphism

Definition

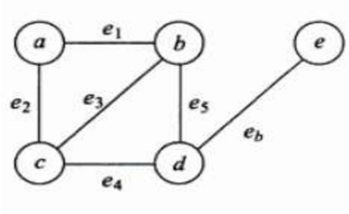
Two graphs G and H are identical (written $G = H$) if $V(G) = V(H)$, $E(G) = E(H)$, and $\psi_G = \psi_H$.

▶ If two graphs are identical then they can clearly be represented by identical diagrams.
▶ It is also possible for two non-identical graphs to be represented by the same diagram. Such graphs are said to be Isomorphic.

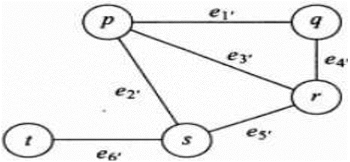
Definition

Two graphs G and H are called isomorphic (denoted by $G \cong H$) to each other if if there are bijections $f : V(G) \to V(H)$ and $\phi : E(G) \to E(H)$ such that
$\psi_G(e) = uv$ if and only if $\psi_H(\phi(e)) = \theta(u)\theta(v)$.
such a pair (θ, ϕ) of mappings is called an isomorphism between G and H.

Are the following two graphs Isomorphic?



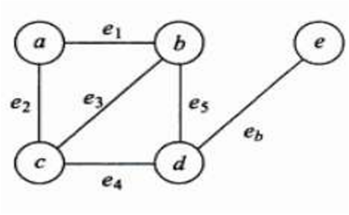
(a) G



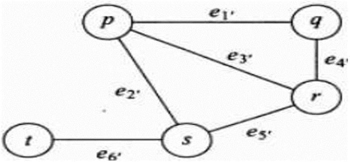
(b) H

- Step 1 : First verify if
  1. $|V(G)| = |V(H)|$ = yes
  2. $|E(G)| = |E(H)|$ = yes
  3. Same degree sequence = (3,3,3,2,1) = yes

Are the following two graphs Isomorphic?



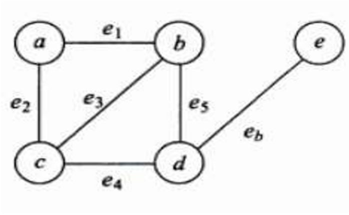
(a) G



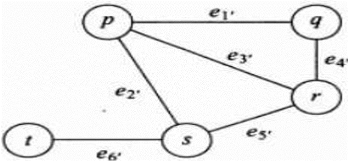
(b) H

- Step 1 : First verify if
  1. $|V(G)| = |V(H)|$ = yes
  2. $|E(G)| = |E(H)|$ = yes
  3. Same degree sequence = (3,3,3,2,1) = yes

Are the following two graphs Isomorphic?



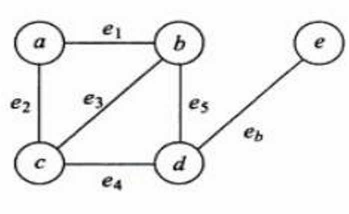
(a) G



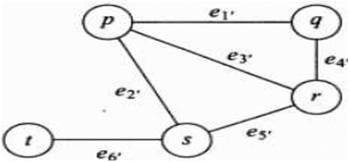
(b) H

- Step 1 : First verify if
  1. $|V(G)| = |V(H)|$ = yes
  2. $|E(G)| = |E(H)|$ = yes
  3. Same degree sequence = (3,3,3,2,1) = yes

Are the following two graphs Isomorphic?



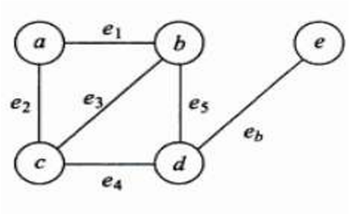
(a) G



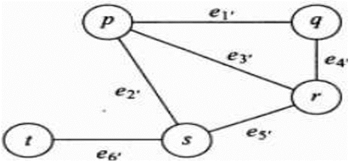
(b) H

- Step 1 : First verify if
  1. $|V(G)| = |V(H)|$ = yes
  2. $|E(G)| = |E(H)|$ = yes
  3. Same degree sequence = (3,3,3,2,1) = yes

Are the following two graphs Isomorphic?



(a) G

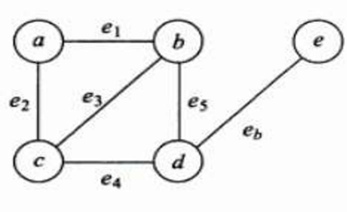


(b) H

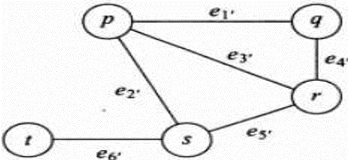
- Step 1 : First verify if
  1. $|V(G)| = |V(H)|$ = yes
  2. $|E(G)| = |E(H)|$ = yes
  3. Same degree sequence = (3,3,3,2,1) = yes

Example

Are the following two graphs Isomorphic?

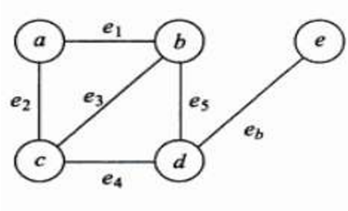


(a) G

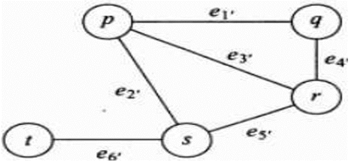


(b) H

- Step 1 : First verify if
  1. $|V(G)| = |V(H)|$ = yes
  2. $|E(G)| = |E(H)|$ = yes
  3. Same degree sequence = (3,3,3,2,1) = yes



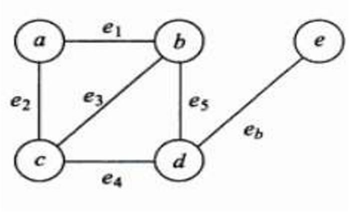
(a) G



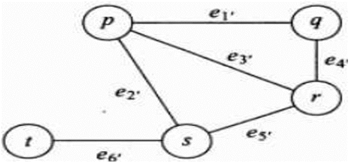
(b) H

- Step 2 : Find bijection between vertices such that their vertex degrees are same.

$$\begin{array}{ccccc} a & b & c & d & e \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow \\ q & p & r & s & t \end{array}$$



(a) G

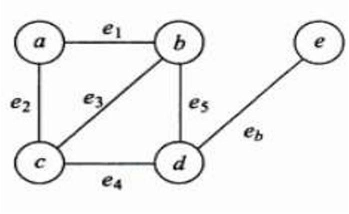


(b) H

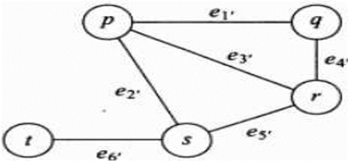
- Step 2 : Find bijection between vertices such that their vertex degrees are same.

$$a \quad b \qquad\qquad c \quad d \qquad\qquad e$$
$$\updownarrow \quad \updownarrow \qquad\qquad \updownarrow \quad \updownarrow \qquad\qquad \updownarrow$$
$$q \quad p \qquad\qquad r \quad s \qquad\qquad t$$

Example



(a) G

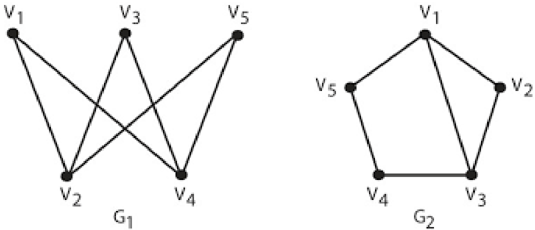


(b) H

- Step 2 : Find bijection between edges based on the vertex mapping,

$$e_1 \leftrightarrow e'_1,\ e_2 \leftrightarrow e'_4,\ e_3 \leftrightarrow e'_3,\ e_4 \leftrightarrow e'_5,\ e_5 \leftrightarrow e'_2,\ e_6 \leftrightarrow e'_6$$

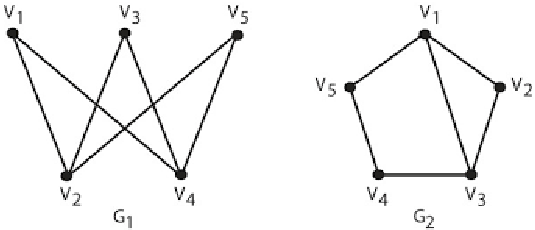
- Are the two graphs isomorphic?



- NO : G_1 is bipartite but G_2 is not.

Example-Isomorphism

- Are the two graphs isomorphic?



- NO : G_1 is bipartite but G_2 is not.

Isomorpism-Check points

1. Count the vertices. The graphs must have an equal number.
2. Count the edges. The graphs must have an equal number.
3. Check vertex degree sequence. Each graph must have the same degree sequence.
4. Check induced subgraphs for isomorphism. If the subgraphs are not isomorphic, then the larger graphs are not either.
5. Count numbers of cycles/cliques.