# Introducing Myself

- LOW, Kian Hsiang (Bryan)
- Graduate: Carnegie Mellon University
- Undergraduate: SoC, NUS
- Website: http://www.cs.cmu.edu/~bryanlow
- Email: lowkh@comp.nus.edu.sg
- Lecture topics: graphs, trees, sequences, counting, cardinality

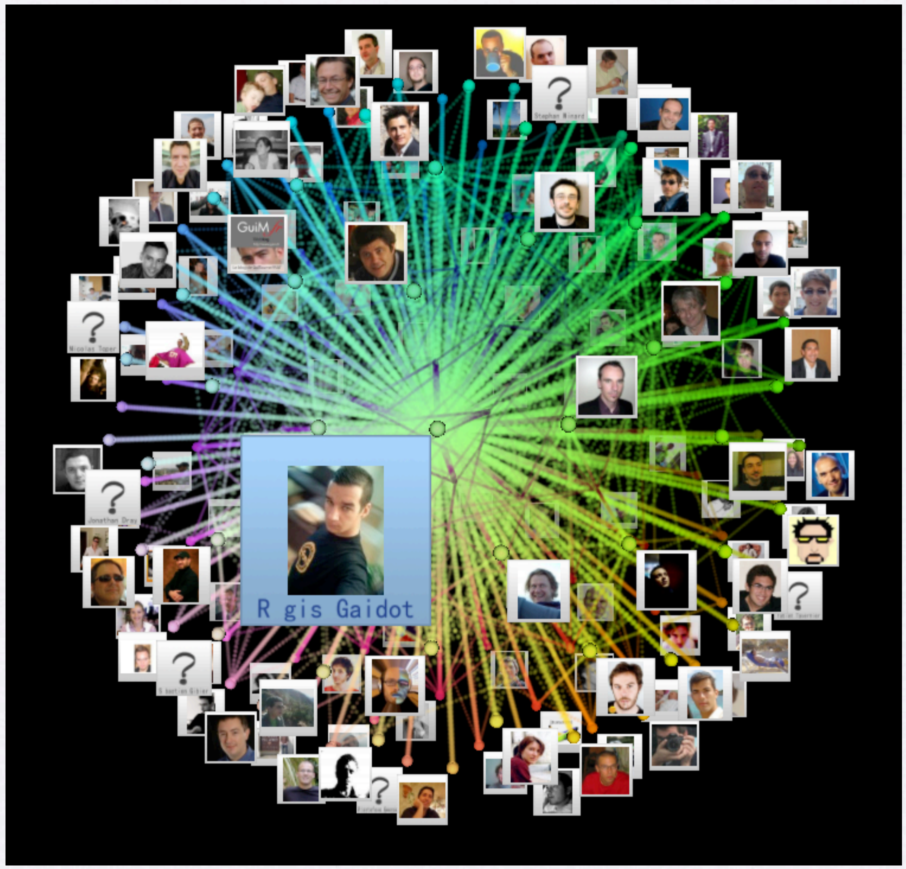
# Graph Theory

Read Rosen, 9.1 - 9.5

# Why Study Graph Theory?

- Not about drawing a curve in x-y plane!
- Model a problem
- Wide variety of real-world applications spanning multiple disciplines: computer network (e.g., LAN), social network (e.g., Facebook), sensor network (e.g., ecological sensing), robotics (e.g., exploration), AI in games (e.g., path planning), multimedia (e.g., NLP, computer vision)
- Advantages of using graph models: useful structural properties, efficient algorithms for solving graph problems

# Graph Model: Facebook



# Topics

- Terminology (Rosen, 9.1 - 9.2)
- Connectivity (Rosen, 9.4)
- Euler tours and Hamilton cycles (Rosen, 9.5)
- Graph representation (Rosen, 9.3)

# Terminology

# Undirected Graphs

**Definitions.**
A **pseudograph** $G = (V(G), E(G), f_G)$ consists of

- a nonempty **vertex set** $V(G)$ of vertices (or nodes),
- an **edge set** $E(G)$ of edges, and
- an incidence function $f_G : E(G) \to \{\{u, v\} \mid u, v \in V(G)\}$.

In an undirected graph, the edges are said to be **undirected**.

An edge $e$ is called a **loop** if $f_G(e) = \{u, u\} = \{u\}$ for some $u \in V(G)$.

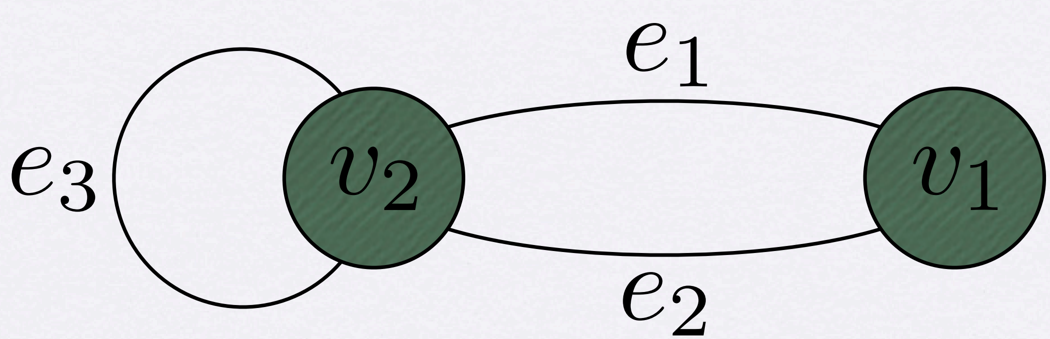
Two distinct edges $e_1$ and $e_2$ are called **multiple** or **parallel edges** if $f_G(e_1) = f_G(e_2)$.

# Undirected Graphs

**Example (Pseudograph).** $H = (V(H), E(H), f_H)$ where

- $V(H) = \{v_1, v_2\}$,
- $E(H) = \{e_1, e_2, e_3\}$, and
- $f_H$ is defined by $f_H(e_1) = \{v_1, v_2\}$, $f_H(e_2) = \{v_1, v_2\}$, $f_H(e_3) = \{v_2, v_2\}$ .

Note that $e_1$ and $e_2$ are parallel edges and $e_3$ is a loop.



# Undirected Graphs

**Definitions.**
A **simple graph** is a pseudograph with no loops and no parallel edges.

A **multigraph** is a pseudograph with no loops.

An edge $e$ is said to be **incident with** vertices $u$ and $v$ (and vice versa) if $f_G(e) = \{u, v\}$. The edge $e$ is said to **connect** its **endpoints** $u$ and $v$.

Two vertices are called **adjacent** (or **neighbors**) if they are incident with a common edge.

Two edges are called **adjacent** if they are incident with a common vertex.

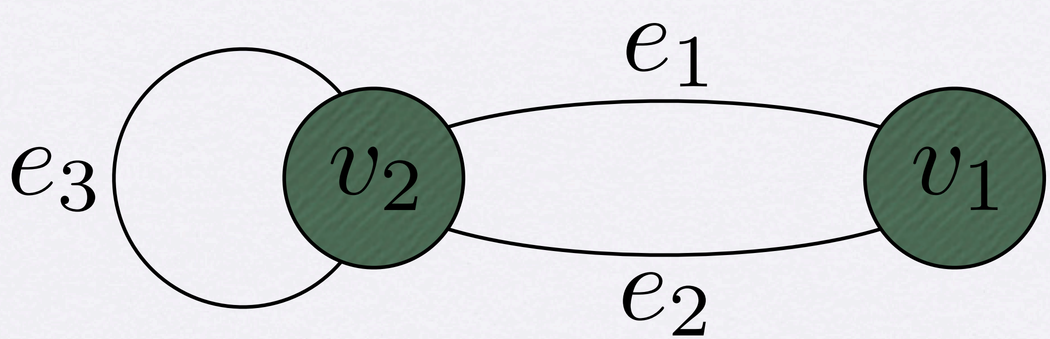
# Undirected Graphs

**Example (Graph *H*).**
Vertex $v_2$ is incident with edges $e_1, e_2, e_3$.

Edge $e_1$ is incident with and connects vertices $v_1, v_2$.

Vertices $v_1$ and $v_2$ are adjacent.

Edges $e_1$ and $e_2$ are adjacent.



# Directed Graphs or Digraphs

**Definitions.**
A **directed multigraph** $D = (V(D), E(D), f_D)$ consists of

- a nonempty **vertex set** $V(D)$ of vertices,
- an **edge set** $E(D)$ of directed edges (or arcs), and
- an incidence function $f_D : E(D) \to \{(u, v) \mid u, v \in V(D)\}$.

The directed edge $e$ is said to **start** at **initial vertex** $u$ and **end** at **terminal** or **end vertex** $v$ if $f_D(e) = (u, v)$.

A directed edge $e$ is called a **loop** if $f_D(e) = (u, u)$ for some $u \in V(D)$.

# Directed Graphs

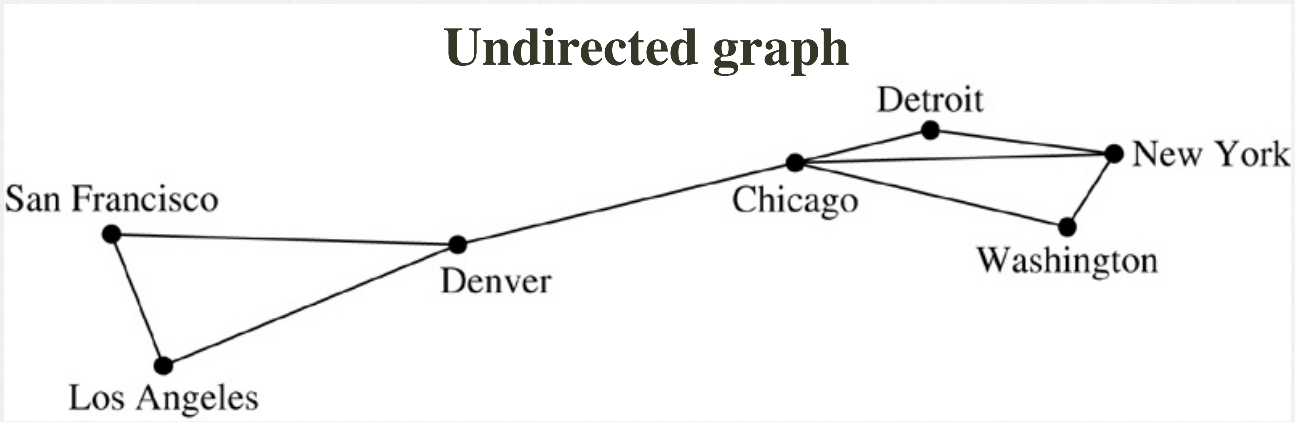
**Definitions.**
The directed edges $e_1$ and $e_2$ are called **multiple directed edges** if $f_D(e_1) = f_D(e_2)$.

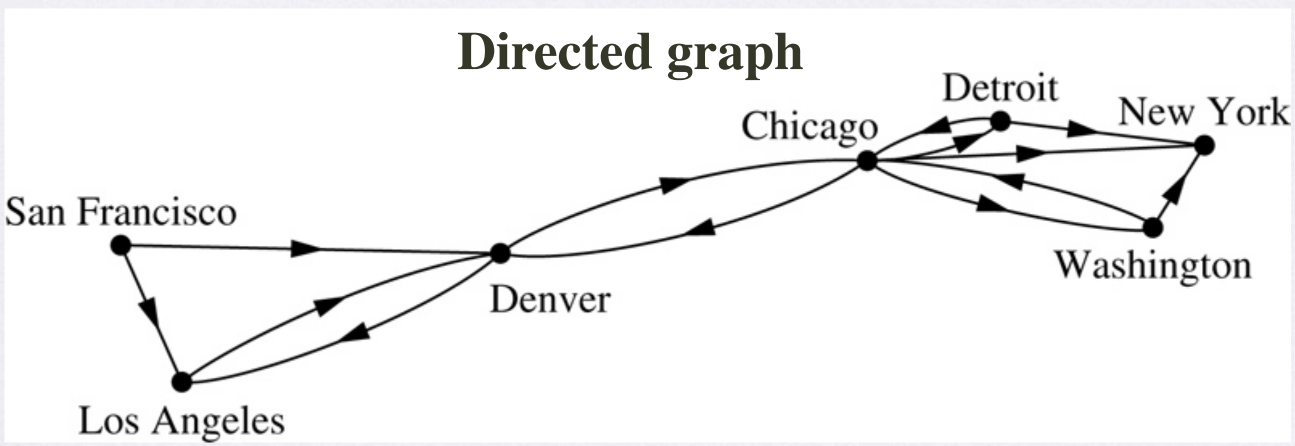
A **simple directed graph** is a directed multigraph with no loops and no multiple directed edges.

When building a graph model, you have to decide whether

- edges are directed or undirected?
- loops are needed?
- multiple edges are required?

# Examples

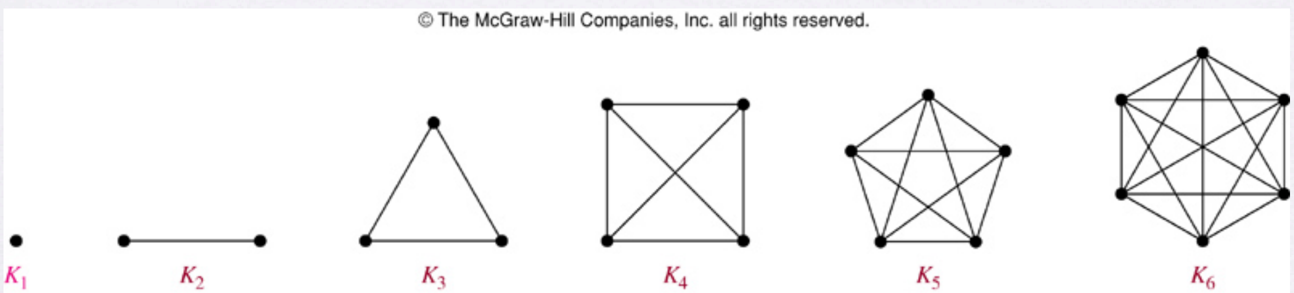

Undirected graph
Detroit
New York
San Francisco
Chicago
Washington
Denver
Los Angeles


Directed graph
Detroit
New York
Chicago
San Francisco
Washington
Denver
Los Angeles

# Complete Graphs

**Definition.** A **complete graph on *n* vertices**, denoted by $K_n$, is a simple graph in which every two distinct vertices are adjacent.

Note that $|E(K_n)| = \binom{n}{2}$.




The intersection of edges are NOT vertices!

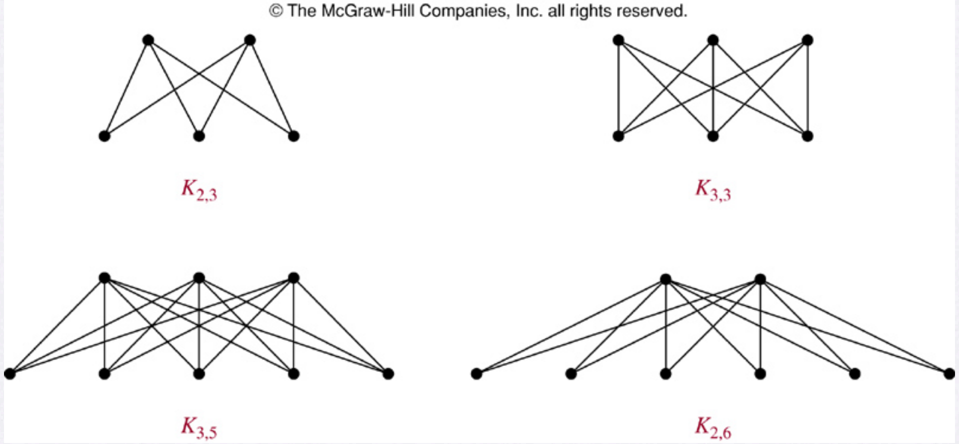
# Complete Bipartite Graphs

**Definition.** A **complete bipartite graph on (*m,n*) vertices**, denoted by $K_{m,n}$, is a simple graph with

- $V(K_{m,n}) = \{u_1, \ldots, u_m\} \cup \{ v_1, \ldots, v_n\}$, and
- $E(K_{m,n}) = \{\{u_i, v_j\} \mid i = 1, \ldots, m; j = 1, \ldots, n\}$ .

Note that $|E(K_{m,n})| = m \times n$ .



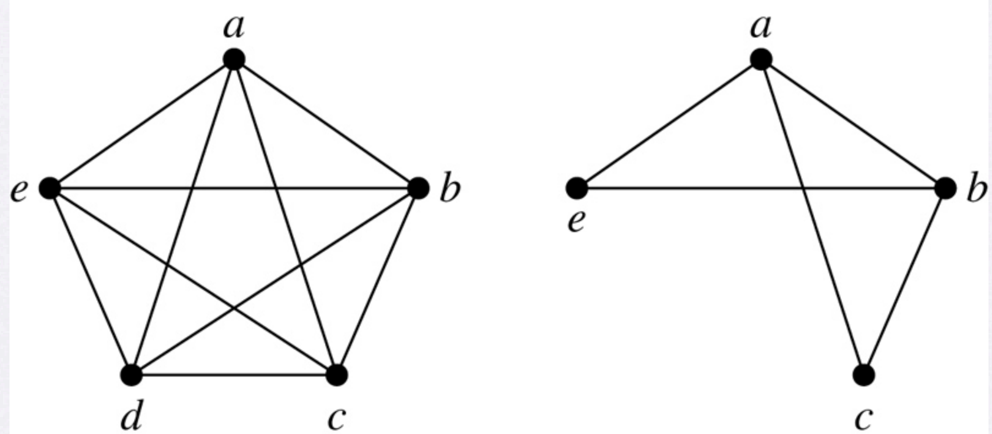


# Subgraphs

**Definition.** A graph $H$ is called a **subgraph** of a graph $G$ if

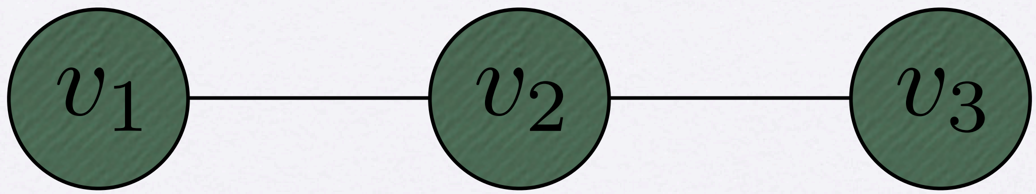
- $V(H) \subseteq V(G)$,
- $E(H) \subseteq E(G)$, and
- $\forall e \in E(H)\ (f_H(e) = f_G(e))$.

Subgraphs of $K_5$



# Subgraphs

**Example.**




- The simple graph above has 12 subgraphs
- Subgraphs without edges = $2^3 - 1 = 7$
- Subgraphs with 1 edge = 4
- Subgraphs with 2 edges = 1

# Vertex Degree

**Definitions.**
The **degree** $d_G(v)$ of a vertex $v$ in an undirected graph $G$ is the number of edges incident with $v$, each loop counting as two edges.

A vertex of degree zero is called **isolated**.

# The Handshake Theorem

**Theorem 1 (The Handshake Theorem).** Let $G$ be an undirected graph. Then,

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

**Corollary 1.** Sum of degrees of all vertices of $G$ is even.

*Example.* Can each person in a social network of 9 people have exactly 5 friends? No!

# The Handshake Theorem

**Theorem 1 (The Handshake Theorem).** Let $G$ be an undirected graph. Then,

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

*Proof.*

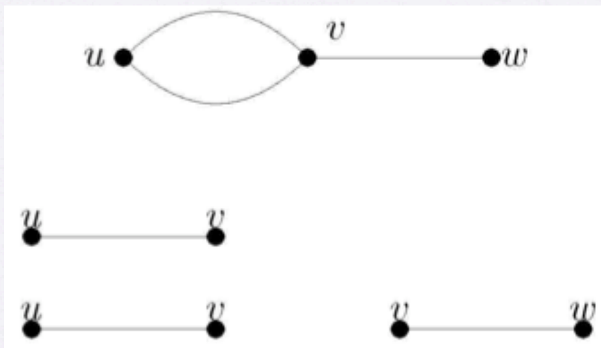
1. List each edge and label its endpoints with the vertex names.
2. The number of times each vertex name is used is the vertex degree.
3. All vertex names appear $2|E(G)|$ times, by 1.
4. This is also the sum of all the vertex degrees, by 2.

# The Handshake Theorem

**Proof Illustration.**

Graph G:

Edges:




Then,

$$d_G(u) + d_G(v) + d_G(w) = 2 + 3 + 1 = 6 = 2 \times |E(G)| \ .$$

# Vertex Degree

**Corollary 2.** In an undirected graph, the number of vertices of odd degree is even.

*Proof.*

1. Let $V_1$ and $V_2$ be the sets of vertices of odd and even degree in G, respectively. Then,

$$\sum_{v \in V_1} d_G(v) + \sum_{v \in V_2} d_G(v) = \sum_{v \in V(G)} d_G(v) \ .$$

2. RHS is even, by Theorem 1. So, LHS is even.

3. $\sum_{v \in V_1} d_G(v)$ is even.

4. Since $\sum_{v \in V_1} d_G(v) = \sum_{v \in V_1} (d_G(v) - 1) + \sum_{v \in V_1} 1$ and $|V_1| = \sum_{v \in V_1} 1$, $|V_1|$ is even.

# Applying Corollary 2

Is there an undirected graph with 10 vertices of degrees 1, 1, 2, 2, 2, 3, 4, 4, 4, and 6?

No, there are 3 vertices with odd degrees!

# Connectivity

# Walks, Trails, and Paths

**Definitions.**
A **walk** of length $n$ in an undirected graph $G$ is a finite alternating sequence of vertices and edges of $G$,

$$v_0\ e_1\ v_1\ e_2 \dots v_{n-1}\ e_n\ v_n\ ,$$

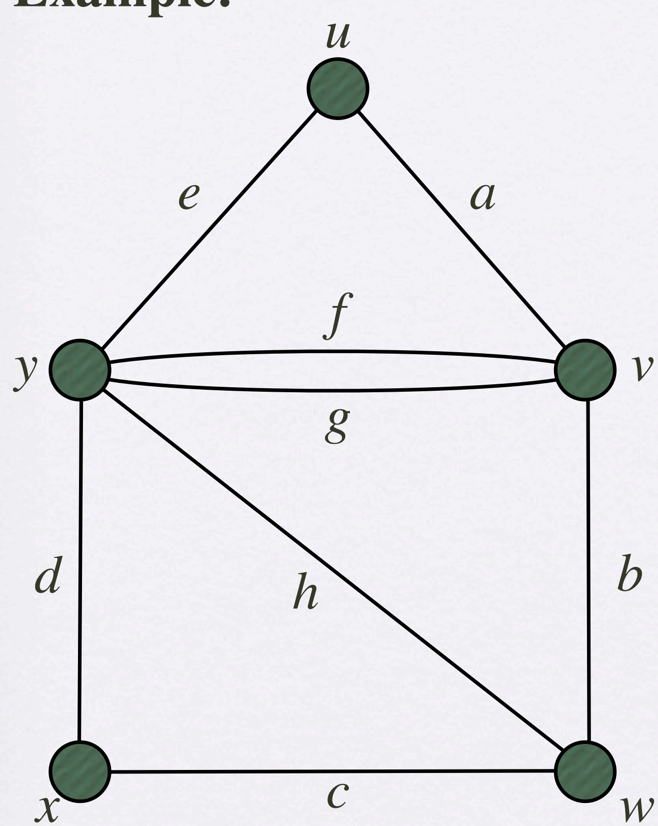
where $e_i$ connects its endpoints $v_{i-1}$ and $v_i$ for $i = 1, \dots, n$. Vertices $v_0$ and $v_n$ are called the **origin** and **terminus**, respectively, and $v_1, \dots, v_{n-1}$ are called **internal vertices**. In a simple graph, a walk can be specified uniquely by its vertex sequence $v_0\ v_1 \dots v_n$.

A **trail** is a walk with distinct edges.

A **path** is a trail with distinct vertices.

# Walks, Trails, and Paths

**Example.**




Walk: *uavfyfvgyhwbv*

Trail: *wcxdyhwbvgy*

Path: *xcwhyeuav*

# Connectedness

**Definition.** An undirected graph $G$ is called **connected** if there is a walk between every pair of distinct vertices of $G$.

**Theorem 2.** There is a path between every pair of distinct vertices of a connected undirected graph.

*Proof.*

1. Let $u$ and $v$ be two distinct vertices of $G$. There is at least one walk between $u$ and $v$, by definition.

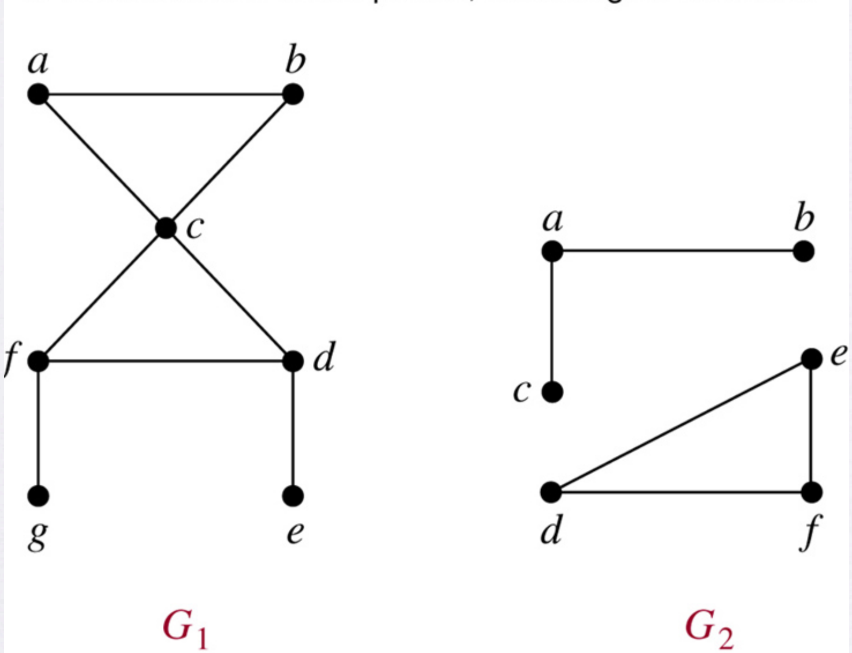
2. Choose the walk of least length.

3. This walk of least length is a path.

# Connectedness

**Example.**






Graph $G_1$ is connected but graph $G_2$ is not.

# Connected Components

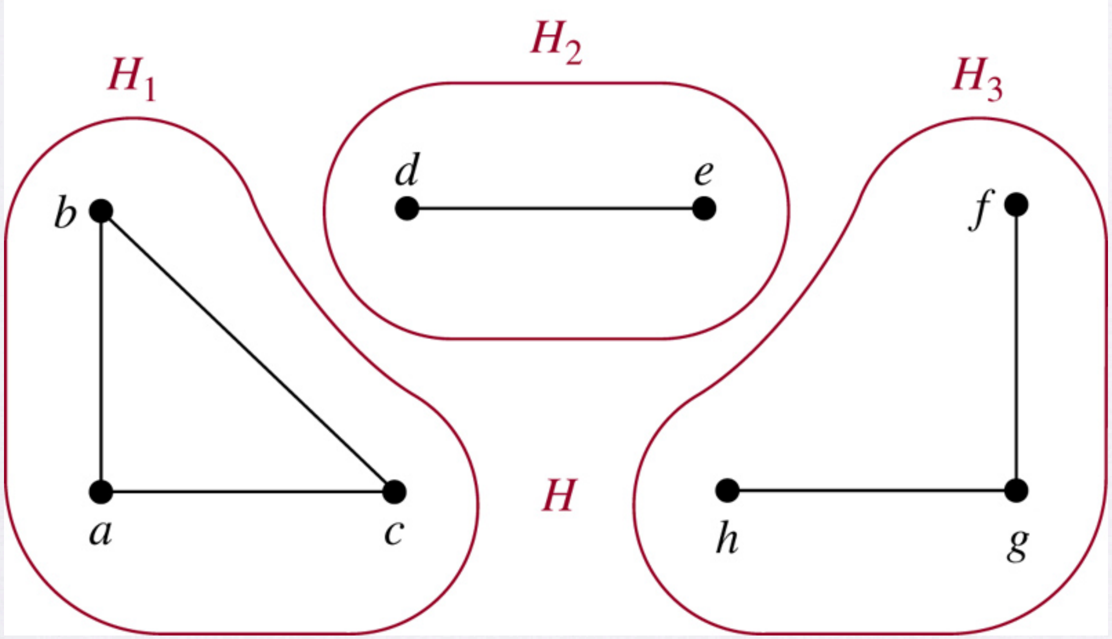
**Definition.** An undirected graph $H$ is called a **connected component** of the undirected graph $G$ if

1. $H$ is a subgraph of $G$,
2. $H$ is connected, and
3. if another connected subgraph $G'$ of $G$ has $H$ as a subgraph, $G' = H$.

# Connected Components

**Example.**



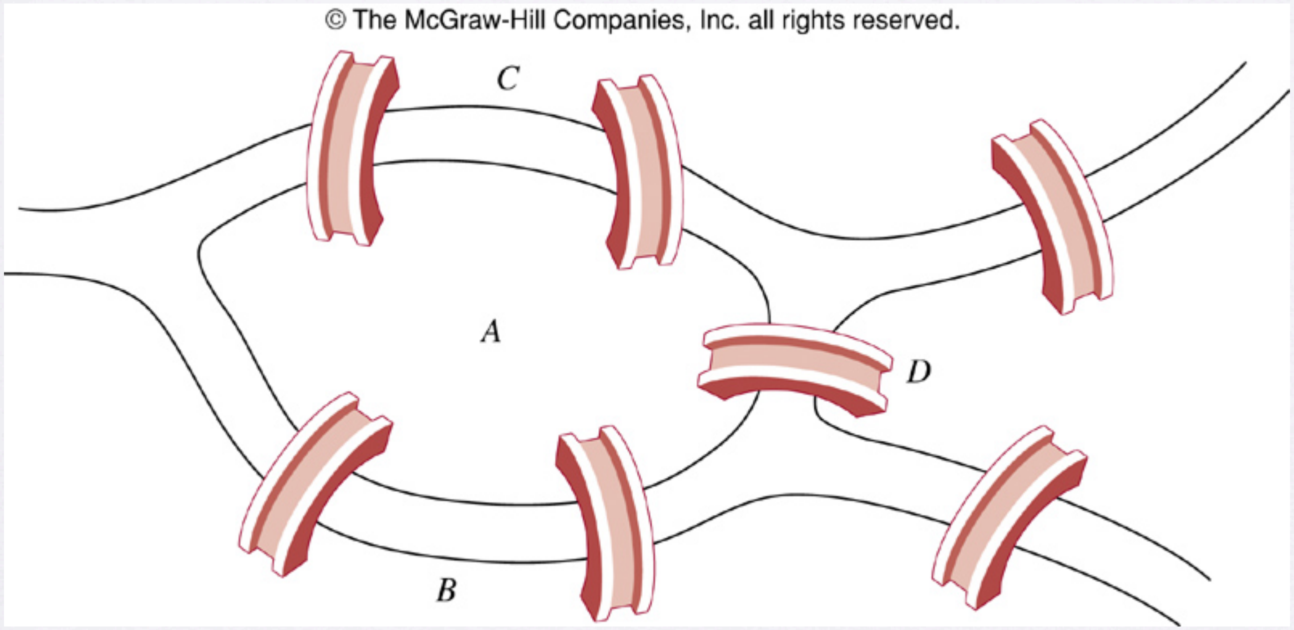



Subgraphs $H_1$, $H_2$, and $H_3$ are connected components of $H$.

# Euler Tours

# Euler Tours

**Example.**




Königsburg and its 7 bridges can be modeled using a multigraph.

# Euler Trails and Tours

**Definitions.**
An **Euler trail** in $G$ is a trail traversing every edge of $G$.

A **closed walk** is a walk with positive length that starts and ends at the same vertex (i.e., same origin and terminus).

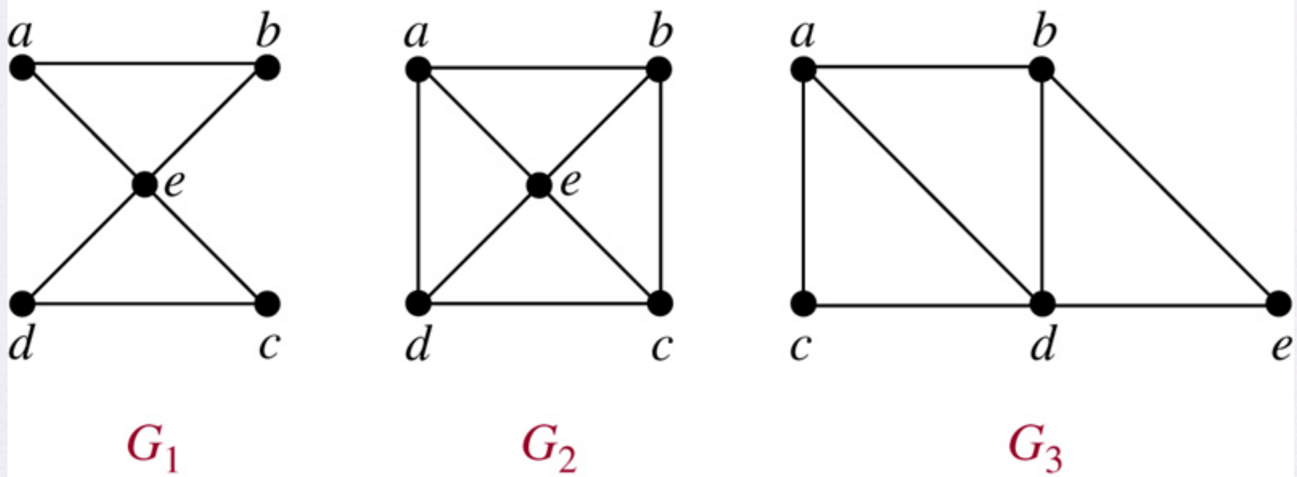
A **tour** in $G$ is a closed walk that traverses each edge of $G$ at least once.

An **Euler tour** in $G$ is a tour traversing each edge *exactly* once (i.e., closed Euler trail).

# Euler Trails and Tours

**Example.**






Only $G_1$ has an Euler tour *aecdeba*.

Out of $G_2$ and $G_3$, only $G_3$ has an Euler trail *acdebdab*.

# Euler Tours

**Theorem 3.** If a nonempty connected multigraph has an Euler tour, it has no vertices of odd degree.

*Proof.*

1. Let $G$ be the graph with an Euler tour $C$ starting and ending at vertex $u$.

2. Each time a vertex $v$ occurs as an internal vertex of $C$, 2 of its edges are accounted for.

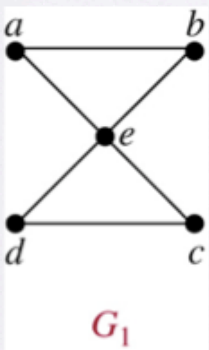
3. Since $C$ contains every edge of $G$, $d_G(v)$ is even for all $v \neq u$.

4. Since $C$ starts and ends at $u$, $d_G(u)$ is even.

# Euler Tours

**Theorem 3.** If a nonempty connected multigraph has an Euler tour, it has no vertices of odd degree.

**Example.**



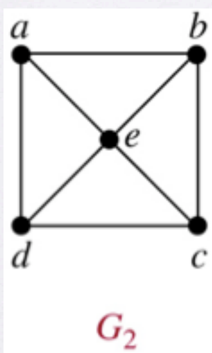

Recall $G_1$ has an Euler tour *aecdeba*.

All vertices have even degree.

# Euler Tours

**Contrapositive of Theorem 3.** If some vertex of a nonempty connected multigraph has odd degree, it does not have an Euler tour.

**Example.**




All vertices except vertex $e$ have odd degree.

$G_2$ does not have an Euler tour, by Theorem 3.

# Euler Tours

**Theorem 4.** If a nonempty connected multigraph has no vertices of odd degree, it has an Euler tour.

*Proof.*

1. Pick any vertex $u$ of graph $G$ to start.
2. Since every vertex has even degree, a closed trail $C$ can be chosen from $G$.
3. If $C$ contains every edge of $G$, it is an Euler tour of $G$.
4. Otherwise, construct subgraph $G'$ by removing all edges of $C$ from $G$ and any resulting isolated vertices. $G'$ may be disconnected but every vertex of $G'$ has even degree.

# Euler Tours

**Theorem 4.** If a nonempty connected multigraph has no vertices of odd degree, it has an Euler tour.

*Proof.*

5. Since $G$ is connected, a vertex $u'$ common to $C$ and $G'$ can be picked.

6. Starting at $u'$, choose a closed trail $C'$ from $G'$.

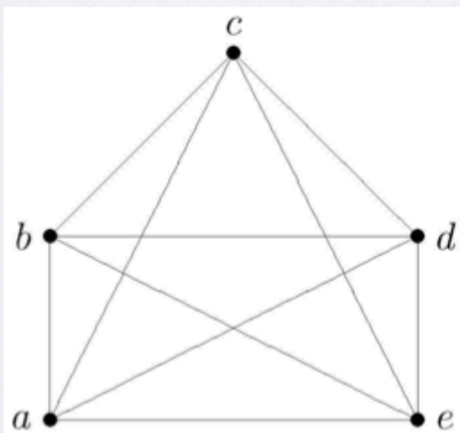
7. Patch $C$ and $C'$ together into one closed trail $C''$.

8. Let $C = C''$ and go back to Step 3.

# Euler Tours

**Theorem 4.** If a nonempty connected multigraph has no vertices of odd degree, it has an Euler tour.

**Example.**



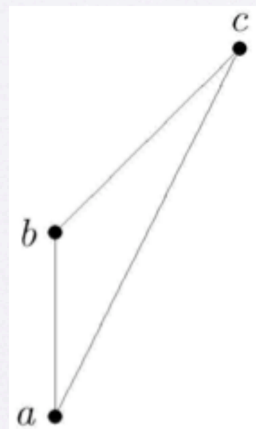

Consider $K_5$.

For any vertex $v$ of $K_5$, $d_G(v) = 4$.

Since every vertex of $K_5$ has even degree, $K_5$ has an Euler tour, by Theorem 4.

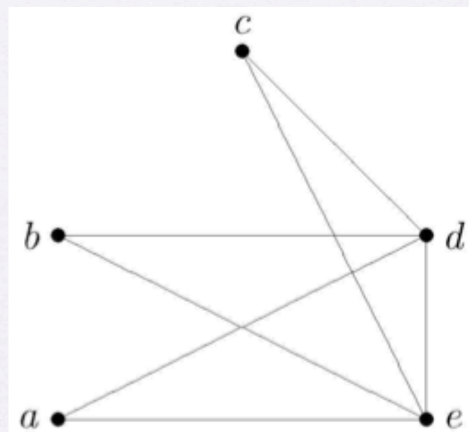
# Constructing Euler Tour

First, pick vertex $a$ of $K_5$. Then, choose a closed trail $abca$ from $K_5$ starting at $a$.



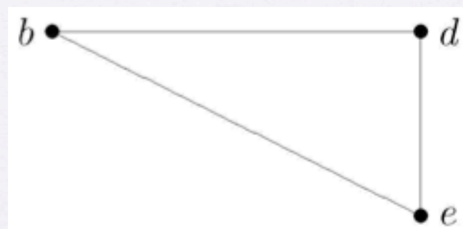
# Constructing Euler Tour

Construct the following subgraph by removing all edges of the closed trail *abca* from $K_5$ and any resulting isolated vertices:



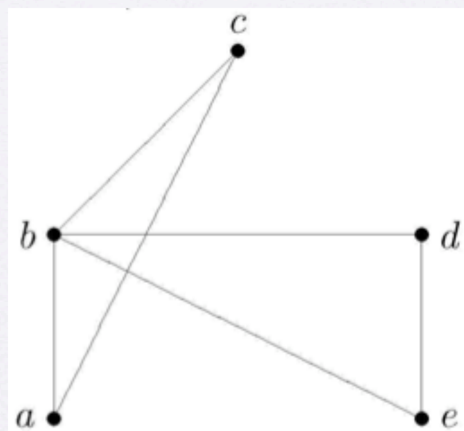
# Constructing Euler Tour

Pick vertex *b* common to the closed trail *abca* and the subgraph. Choose a closed trail *bedb* from the subgraph starting from *b*.



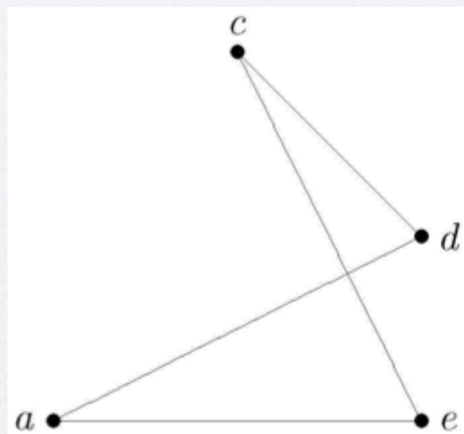
# Constructing Euler Tour

Patch the closed trail *abca* and the closed trail *bedb* together into one closed trail *abedbca*.



# Constructing Euler Tour

Construct the following subgraph by removing all edges of the closed trail *abedbca* from $K_5$ and any resulting isolated vertices:



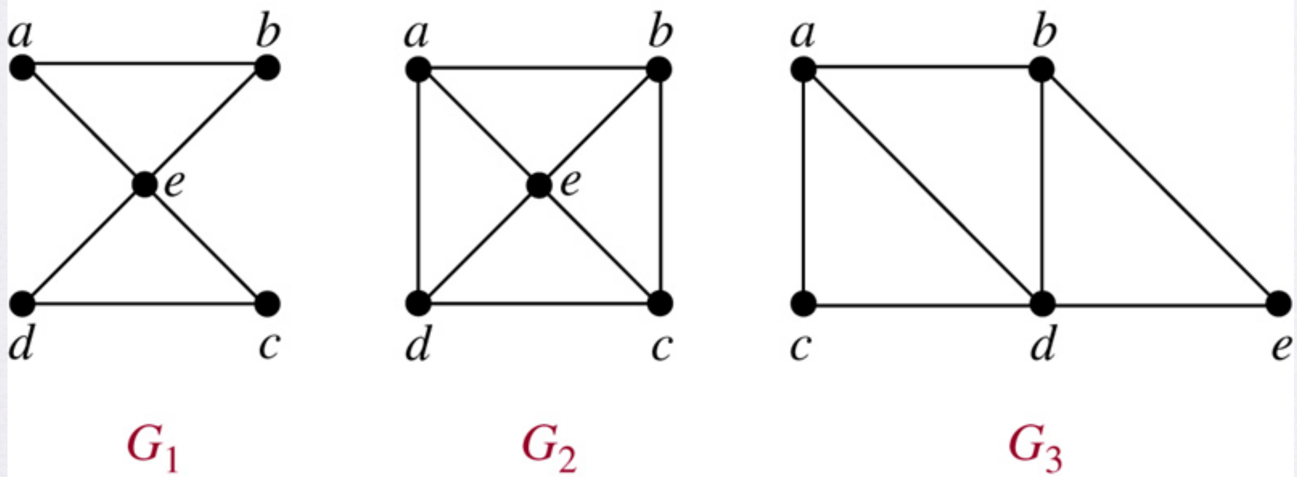

Pick vertex *a* common to the closed trail *abedbca* and the subgraph. Choose a closed trail *adcea* from the subgraph starting from *a*. Patch the closed trail *abedbca* and the closed trail *adcea* together into an Euler tour *abedbcadcea*.

# Euler Tours

**Theorem 4.** If a nonempty connected multigraph has no vertices of odd degree, it has an Euler tour.

**Example.**






Only $G_1$ has no vertices of odd degree.
So, $G_1$ has an Euler tour *aecdeba*, by Theorem 4.

# Euler Trails and Tours

**Theorem 5.** A nonempty connected multigraph has an Euler tour if and only if it has no vertices of odd degree.

*Proof.*
Theorems $3 + 4$.

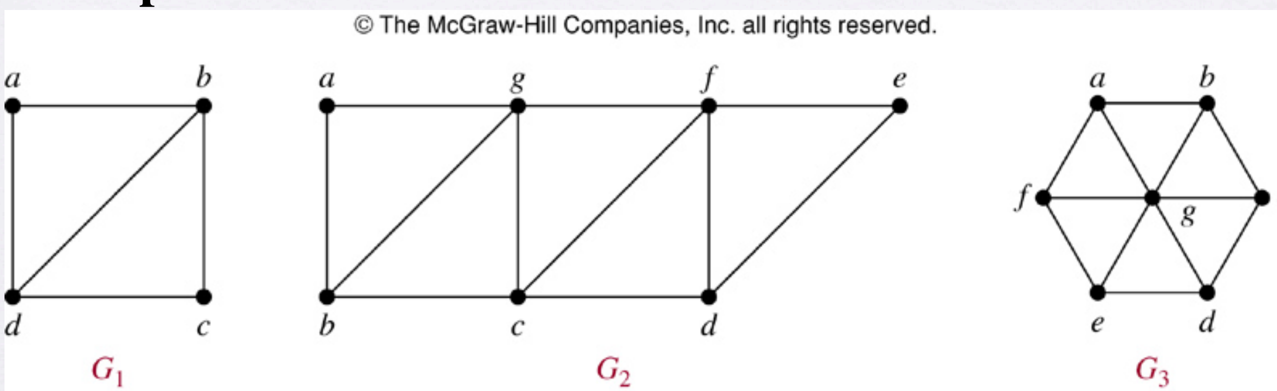
**Corollary 3.** A nonempty connected multigraph has an Euler trail but not an Euler tour if and only if it has exactly two vertices of odd degree.

*Proof.*
Omitted.

# Euler Trails

**Corollary 3.** A nonempty connected multigraph has an Euler trail but not an Euler tour if and only if it has exactly two vertices of odd degree.

**Example.**




$G_1$ and $G_2$ have exactly 2 vertices of odd degree, namely, $b$ and $d$. So, by Corollary 3, $G_1$ and $G_2$ have Euler trails *dabcdb* and *bagfedcgbcfd*, respectively. $G_3$ has 6 vertices of odd degree. So, $G_3$ has no Euler trail, by Corollary 3.

# Hamilton Cycles

# Hamilton Paths and Cycles

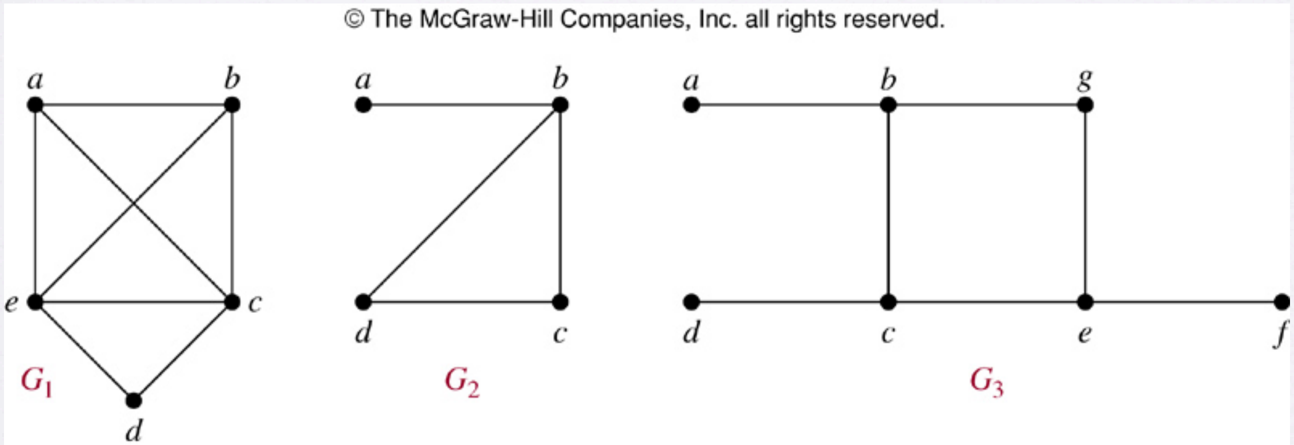
**Definitions.**
A **Hamilton path** in $G$ is a path containing every vertex of $G$.

A **cycle** in $G$ is a closed trail whose origin and internal vertices are distinct.

A **Hamilton cycle** in $G$ is a cycle containing every vertex of $G$.

# Hamilton Paths and Cycles

**Example.**




$G_1$ has a Hamilton cycle *abcdea*.

$G_2$ does not have a Hamilton cycle but has a Hamilton path *abcd*.

$G_3$ does not have a Hamilton cycle or Hamilton path.

# Hamilton Cycle

Recall the necessary and sufficient condition for a graph to have a Euler tour (Theorem 5).

No nontrivial necessary and sufficient condition for a graph to have a Hamilton cycle!

# Hamilton Cycles

**Proposition 1.** If a multigraph $G$ has a Hamilton cycle, $G$ has a subgraph $H$ with the following properties:

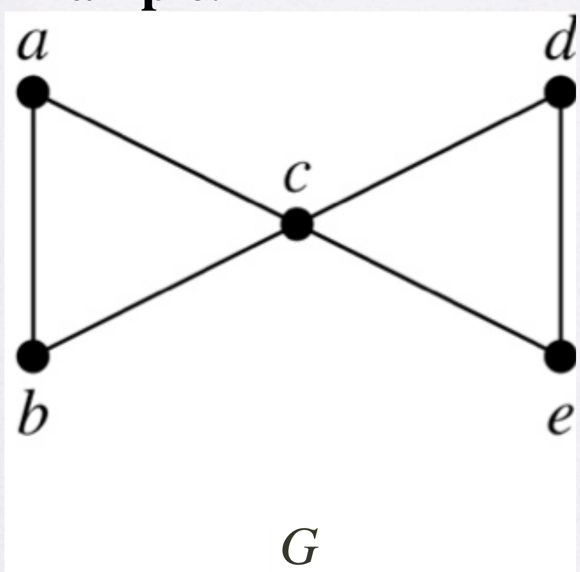
1. $V(H) = V(G)$,
2. $H$ is connected,
3. $|E(H)| = |V(H)|$, and
4. $\forall v \in V(H)\ d_H(v) = 2$ .

*Proof.*
Omitted.

# Hamilton Cycles

**Contrapositive of Proposition 1.** If a multigraph $G$ does not have a subgraph $H$ with properties 1 to 4, $G$ does not have a Hamilton cycle.

**Example.**




$G$ does not have a Hamilton cycle, by Proposition 1.

# Graph Representation

# Adjacency Matrix

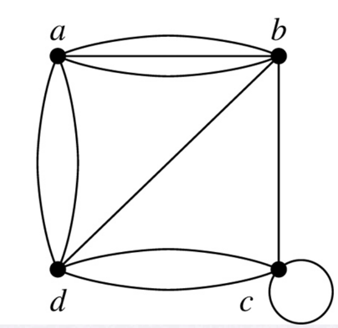
**Definition.** Let $G$ be an undirected graph and the vertices are ordered as $V(G) = \{v_1, \ldots, v_n\}$. The **adjacency matrix** of $G$ is the $n \times n$ matrix $\mathbf{A}(G) = [a_{ij}]$ such that

$$a_{ij} = \left|\{e \in E(G) \mid f_G(e) = \{v_i, v_j\}\}\right| .$$

*Remark.* $\mathbf{A}(G)$ is symmetric (i.e., $a_{ij} = a_{ji}$ for $1 \leq i, j \leq n$).

**Example.**






$$\mathbf{A}(G) = \begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}$$

# Adjacency Matrix

**Definition.** Let $D$ be a directed graph and the vertices are ordered as $V(D) = \{v_1, \ldots, v_n\}$. The **adjacency matrix** of $D$ is the $n \times n$ matrix $\mathbf{A}(D) = [a_{ij}]$ such that

$$a_{ij} = \left|\{e \in E(D) \mid f_D(e) = (v_i, v_j)\}\right| .$$

*Remark.* $\mathbf{A}(D)$ may not be symmetric.

# Counting Walks Between Vertices

**Theorem 6.** Let $G$ be a graph with vertices ordered as $V(G) = \{v_1, \ldots, v_n\}$ and with adjacency matrix $\mathbf{A}(G)$. Then, for each nonnegative integer $k$, the number of walks of length $k$ from $v_i$ to $v_j$ equals to the $(i, j)$th entry of $\mathbf{A}(G)^k$.

*Remark.* $\mathbf{A}(G)^0 = \mathbf{I}$.

*Proof.*
Omitted.

# Topics Not Covered

- Bipartite graphs (Rosen, 9.2)
- Isomorphism (Rosen, 9.3)
- Paths and connectedness in directed graphs (Rosen, 9.4)
- Representing graphs with incidence matrices (Rosen, 9.3)