# Discrete Mathematical Models

**Lecture 22**

Kane Townsend

Semester 2, 2024

# D: Graph Theory

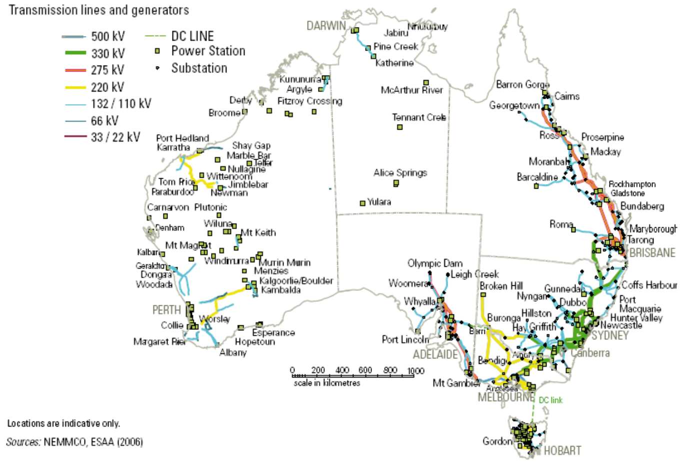
| | |
|---|---|
| Text Reference (Epp) | 3ed: Chapter 11 |
| | 4ed: Chapter 10 |
| | 5ed: Chapter 10 |

These references may not *completely* cover everything in this section, but they do have most it. They also contain a few items we do not cover.

# Motivation

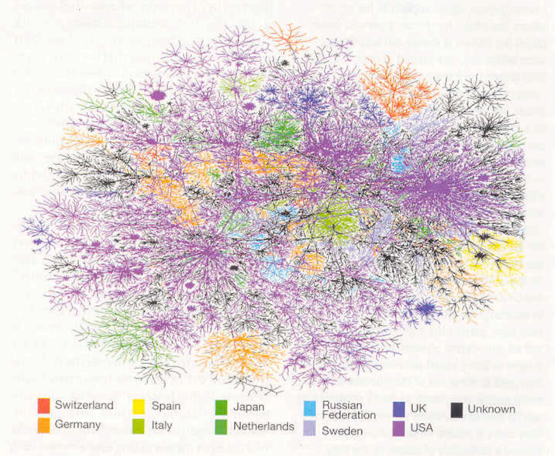
Real-world phenomena often modeled with graphs:

# Australian Power Transmission Network

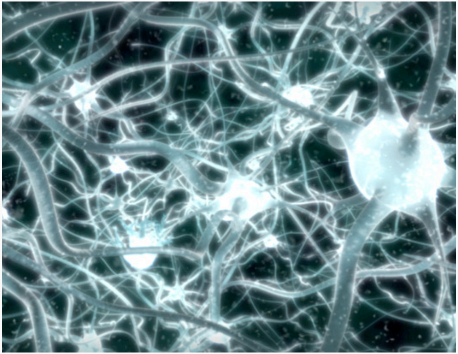


# Complex Network Example: Internet

(William R. Cheswick)



# Brain Network



from documentary, 'Inside the living body'
http://abcnews.go.com/2020/popup?id=3560899

# Basic concepts

# Graphs

A **graph** $G$ is a collection of **vertices** and **edges** where:

- The vertices of $G$ form a set denoted by $V(G)$.
- The edges of $G$ form a (multi)set[1] denoted by $E(G)$ where the elements are size-2 multisets

$$E(G) \subseteq \big\{\{x, y\} \mid x, y \in V(G)\big\}.$$

# Graphs

A **graph** $G$ is a collection of **vertices** and **edges** where:

- The vertices of $G$ form a set denoted by $V(G)$.
- The edges of $G$ form a (multi)set[1] denoted by $E(G)$ where the elements are size-2 multisets

$$E(G) \subseteq \big\{\{x, y\} \mid x, y \in V(G)\big\}.$$

**Comments:**

Each edge is specified by a pair of vertices (can be same vertices) and each edge determines one or two vertices, which are called its **endpoints**.

We only consider **finite** graphs; *i.e.* $V(G)$ and $E(G)$ are both finite sets.

# Graphs

A **graph** $G$ is a collection of **vertices** and **edges** where:

- The vertices of $G$ form a set denoted by $V(G)$.
- The edges of $G$ form a (multi)set[1] denoted by $E(G)$ where the elements are size-2 multisets

$$E(G) \subseteq \big\{\{x, y\} \mid x, y \in V(G)\big\}.$$

**Comments:**

Each edge is specified by a pair of vertices (can be same vertices) and each edge determines one or two vertices, which are called its **endpoints**.

We only consider **finite** graphs; *i.e.* $V(G)$ and $E(G)$ are both finite sets.

# Graphs

A **graph** $G$ is a collection of **vertices** and **edges** where:

- The vertices of $G$ form a set denoted by $V(G)$.
- The edges of $G$ form a (multi)set[1] denoted by $E(G)$ where the elements are size-2 multisets

$$E(G) \subseteq \big\{\{x, y\} \mid x, y \in V(G)\big\}.$$

**Comments:**

Each edge is specified by a pair of vertices (can be same vertices) and each edge determines one or two vertices, which are called its **endpoints**.

We only consider **finite** graphs; *i.e.* $V(G)$ and $E(G)$ are both finite sets.

[1] As explained in Section C1, a ‘multiset’ is just like a set except that it may contain the same element more than once.

# Diagrams of Graphs

To draw a graph we use:

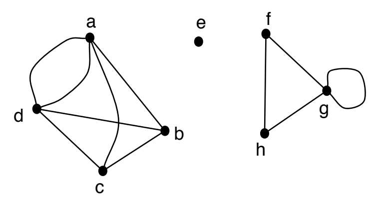
- dots/circles for vertices
- lines for edges

# Diagrams of Graphs

To draw a graph we use:

- dots/circles for vertices
- lines for edges

**Example:** The graph $G$

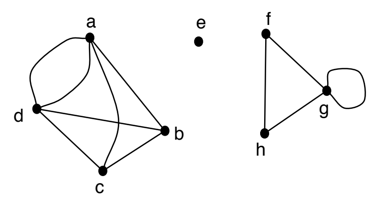


# Diagrams of Graphs

To draw a graph we use:

- dots/circles for vertices
- lines for edges

**Example:** The graph $G$




Has vertex set $V(G) = \{a, b, c, d, e, f, g, h\}$ and edge multiset $E(G) = \{\{a,b\}, \{a,c\}, \{a,d\}, \{a,d\}, \{b,c\}, \{b,d\}, \{c,d\}, \{f,g\}, \{f,h\}, \{g,g\}, \{g,h\}\}$.

# A table of edges

The same graph as a table of labelled edges:

| Edge | Endpoints |
|---|---|
| $e_1$ | $\{a, b\}$ |
| $e_2$ | $\{a, c\}$ |
| $e_3$ | $\{a, d\}$ |
| $e_4$ | $\{a, d\}$ |
| $e_5$ | $\{b, c\}$ |
| $e_6$ | $\{b, d\}$ |

| Edge | Endpoints |
|---|---|
| $e_7$ | $\{c, d\}$ |
| $e_8$ | $\{f, g\}$ |
| $e_9$ | $\{f, h\}$ |
| $e_{10}$ | $\{g, g\}$ |
| $e_{11}$ | $\{g, h\}$ |
| | |

# A table of edges

The same graph as a table of labelled edges:

| Edge | Endpoints |
|---|---|
| $e_1$ | $\{a, b\}$ |
| $e_2$ | $\{a, c\}$ |
| $e_3$ | $\{a, d\}$ |
| $e_4$ | $\{a, d\}$ |
| $e_5$ | $\{b, c\}$ |
| $e_6$ | $\{b, d\}$ |

| Edge | Endpoints |
|---|---|
| $e_7$ | $\{c, d\}$ |
| $e_8$ | $\{f, g\}$ |
| $e_9$ | $\{f, h\}$ |
| $e_{10}$ | $\{g, g\}$ |
| $e_{11}$ | $\{g, h\}$ |
| | |

Notice that $e_3$ is distinct from $e_4$ even though both edges have the same endpoints.

# A vertex adjacency listing

Vertices $u, v \in V(G)$ are **adjacent** if $\{u, v\} \in E(G)$.

- The same graph as a vertex adjacency listing:

| Vertex | Adjacent to: |
|---|---|
| $a$ | $b, c, d, d$ |
| $b$ | $a, c, d$ |
| $c$ | $a, b, d$ |
| $d$ | $a, a, b, c$ |
| $e$ | |
| $f$ | $g, h$ |
| $g$ | $f, g, h$ |
| $h$ | $f, g$ |

# An adjacency matrix

The **adjacency matrix** $(a_{i,j})$ of $G$ is a $|V(G)| \times |V(G)|$ square matrix such that

$$a_{i,j} = \text{ number of edges between vertices } i \text{ and } j$$

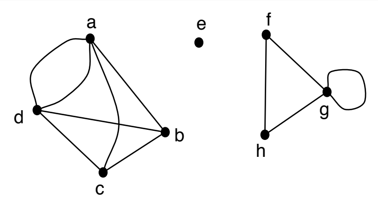
# An adjacency matrix

The **adjacency matrix** $(a_{i,j})$ of $G$ is a $|V(G)| \times |V(G)|$ square matrix such that

$$a_{i,j} = \text{ number of edges between vertices } i \text{ and } j$$

An adjacency matrix for our graph:

$$\begin{array}{c c}
 & \begin{array}{cccccccc} a & b & c & d & e & f & g & h \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{array} &
\begin{bmatrix}
0 & 1 & 1 & 2 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0
\end{bmatrix}
\end{array}$$



## Some Graph Terminology

- An edge connects its **endpoints**.

# Some Graph Terminology

- An edge connects its **endpoints**.
- An edge with both endpoints the same is called a **loop**.

# Some Graph Terminology

- An edge connects its **endpoints**.
- An edge with both endpoints the same is called a **loop**.
- Two edges may connect the same pair of endpoints, in which case they are said to be **parallel**.

# Some Graph Terminology

- An edge connects its **endpoints**.
- An edge with both endpoints the same is called a **loop**.
- Two edges may connect the same pair of endpoints, in which case they are said to be **parallel**.
- Two vertices are **adjacent** if they are connected by an edge; two edges are **adjacent** if they share an endpoint.

## Some Graph Terminology

- An edge connects its **endpoints**.
- An edge with both endpoints the same is called a **loop**.
- Two edges may connect the same pair of endpoints, in which case they are said to be **parallel**.
- Two vertices are **adjacent** if they are connected by an edge; two edges are **adjacent** if they share an endpoint.
- An edge is **incident on** its endpoints.

# Some Graph Terminology

- An edge connects its **endpoints**.
- An edge with both endpoints the same is called a **loop**.
- Two edges may connect the same pair of endpoints, in which case they are said to be **parallel**.
- Two vertices are **adjacent** if they are connected by an edge; two edges are **adjacent** if they share an endpoint.
- An edge is **incident on** its endpoints.
- A vertex with no incident edges is **isolated**.

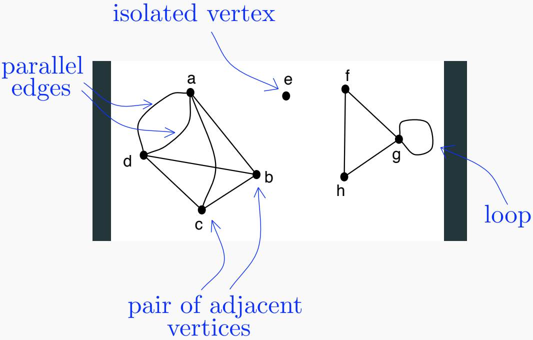
# Some Graph Terminology

- An edge connects its **endpoints**.
- An edge with both endpoints the same is called a **loop**.
- Two edges may connect the same pair of endpoints, in which case they are said to be **parallel**.
- Two vertices are **adjacent** if they are connected by an edge; two edges are **adjacent** if they share an endpoint.
- An edge is **incident on** its endpoints.
- A vertex with no incident edges is **isolated**.
- A graph with no vertices (hence no edges) is **empty**.

# Some Graph Terminology

- An edge connects its **endpoints**.
- An edge with both endpoints the same is called a **loop**.
- Two edges may connect the same pair of endpoints, in which case they are said to be **parallel**.
- Two vertices are **adjacent** if they are connected by an edge; two edges are **adjacent** if they share an endpoint.
- An edge is **incident on** its endpoints.
- A vertex with no incident edges is **isolated**.
- A graph with no vertices (hence no edges) is **empty**.
- The **order** of a graph, $G$, is the number of vertices in it, i.e. $|V(G)|$. (A graph of order '0' is empty.)

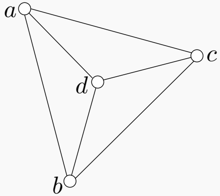
# Some graph concepts illustrated



# Another example

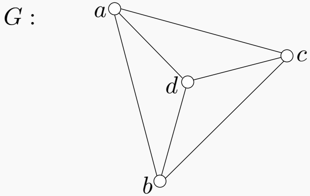
- Tetrahedron Graph

$G:$

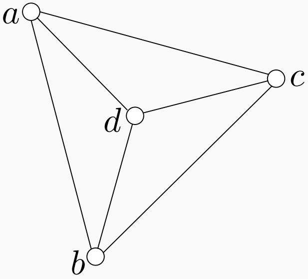


# Another example

- Tetrahedron Graph




  - $V(G) = \{a, b, c, d\}$
  - $E(G) = \big\{\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}\big\}$




- Tetrahedron Graph:

*Adjacency listing:*

| Vertex | Adjacent to: |
|---|---|
| $a$ | $b, c, d$ |
| $b$ | $a, c, d$ |
| $c$ | $a, b, d$ |
| $d$ | $a, b, c$ |

*Adjacency matrix:*

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

# More about graph diagrams

- Position, length, curvedness and orientation in a graph diagram do not matter for the graph represented.

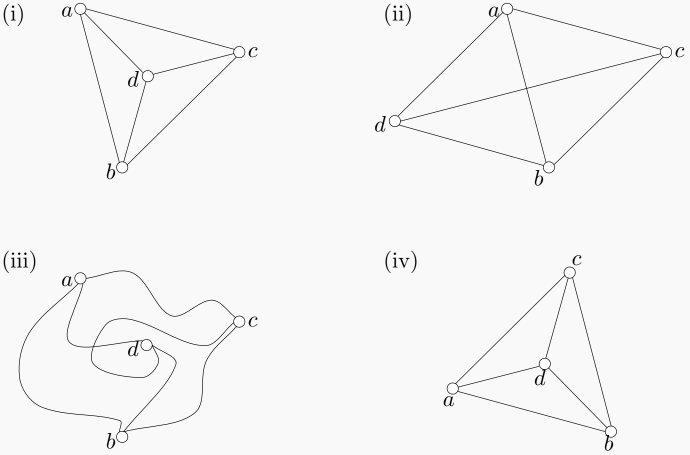
# More about graph diagrams

- Position, length, curvedness and orientation in a graph diagram do not matter for the graph represented.
- The only things which matter are that precisely those vertices in $V(G)$ are shown and precisely those edges in $E(G)$ are shown.

# More about graph diagrams

- Position, length, curvedness and orientation in a graph diagram do not matter for the graph represented.
- The only things which matter are that precisely those vertices in $V(G)$ are shown and precisely those edges in $E(G)$ are shown.
- For instance, the following diagrams all represent the same graph.

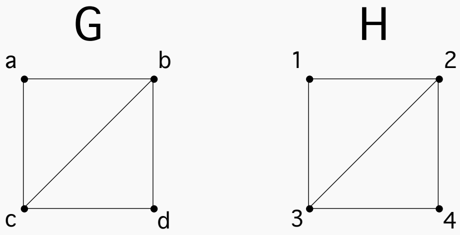
# Four diagrams of the same graph



# Isomorphisms between graphs

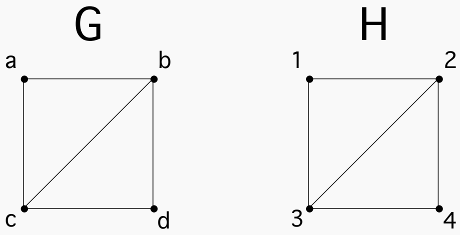
# Isomorphic Graphs

Consider graphs $G$ and $H$ as pictured:



## Isomorphic Graphs

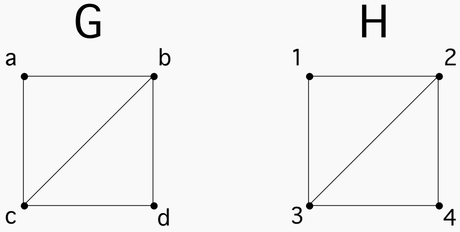
Consider graphs $G$ and $H$ as pictured:




- They are different graphs because their vertex labels are different.

# Isomorphic Graphs

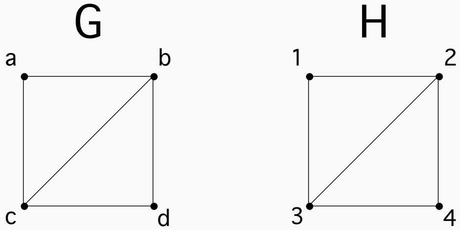
Consider graphs $G$ and $H$ as pictured:




- They are different graphs because their vertex labels are different.
- But they are the same in some sense.

# Isomorphic Graphs

Consider graphs $G$ and $H$ as pictured:




- They are different graphs because their vertex labels are different.
- But they are the same in some sense.
- Formally these graphs are called ‘isomorphic’.

# Isomorphisms

An **isomorphism** between two graphs $G_1$ and $G_2$ is a bijection

$$f : V(G_1) \to V(G_2)$$

such that:

# Isomorphisms

An **isomorphism** between two graphs $G_1$ and $G_2$ is a bijection

$$f : V(G_1) \to V(G_2)$$

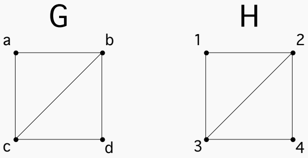
such that:

$\{u, v\}$ is an edge in $E(G_1)$

exactly as many times as

$\{f(u), f(v)\}$ is an edge in $E(G_2)$

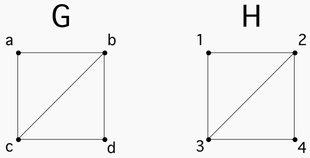
# Isomorphisms




An example of an isomorphism between $G$ and $H$ is the mapping

$$f : V(G) \quad \to \quad V(H)$$

# Isomorphisms

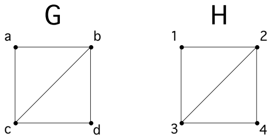



An example of an isomorphism between $G$ and $H$ is the mapping

$$f : V(G) \rightarrow V(H)$$

$$\begin{aligned} a &\mapsto 1 \\ b &\mapsto 2 \\ c &\mapsto 3 \\ d &\mapsto 4 \end{aligned}$$

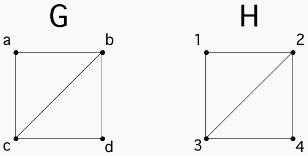
# Isomorphisms




The mapping 'preserves the multiplicity of edges':

| | | | |
|---|---|---|---|
| $\{a, a\}$ | $\mapsto$ | $\{1, 1\}$ | ✓ both have multiplicity 0 |
| $\{a, b\}$ | $\mapsto$ | $\{1, 2\}$ | ✓ both have multiplicity 1 |
| $\{a, c\}$ | $\mapsto$ | $\{1, 3\}$ | ✓ both have multiplicity 1 |
| $\{a, d\}$ | $\mapsto$ | $\{1, 4\}$ | ✓ both have multiplicity 0 |
| $\{b, b\}$ | $\mapsto$ | $\{2, 2\}$ | ✓ both have multiplicity 0 |
| $\{b, c\}$ | $\mapsto$ | $\{2, 3\}$ | ✓ both have multiplicity 1 |
| $\{b, d\}$ | $\mapsto$ | $\{2, 4\}$ | ✓ both have multiplicity 1 |
| $\{c, c\}$ | $\mapsto$ | $\{3, 3\}$ | ✓ both have multiplicity 0 |
| $\{c, d\}$ | $\mapsto$ | $\{3, 4\}$ | ✓ both have multiplicity 1 |
| $\{d, d\}$ | $\mapsto$ | $\{4, 4\}$ | ✓ both have multiplicity 0 |

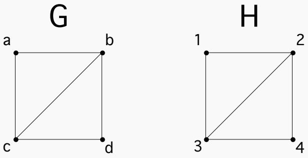
# Isomorphisms




A different example of an isomorphism between $G$ and $H$ is the mapping

$$g : V(G) \quad \rightarrow \quad V(H)$$

# Isomorphisms




A different example of an isomorphism between $G$ and $H$ is the mapping

$$g : V(G) \quad \rightarrow \quad V(H)$$

$$\begin{aligned} a &\mapsto 1 \\ b &\mapsto 3 \\ c &\mapsto 2 \\ d &\mapsto 4 \end{aligned}$$

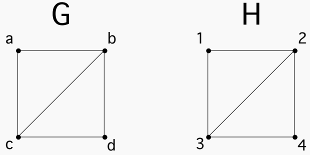
# Isomorphic Graphs

If there exists an isomorphism between two graphs then the graphs are said to be **isomorphic**.

# Isomorphic Graphs

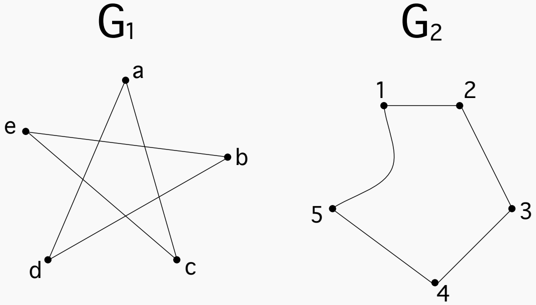
If there exists an isomorphism between two graphs then the graphs are said to be **isomorphic**.

**Example:** Graphs $G$ and $H$ are isomorphic.



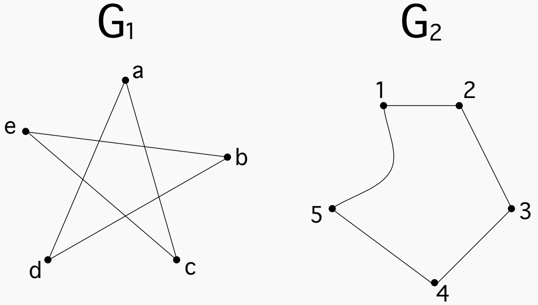
# Isomorphic Graphs

- The following graphs pictured are isomorphic.



# Isomorphic Graphs

- The following graphs pictured are isomorphic.




- Can you specify an explicit isomorphism between them?

# A research aside:

Emeritus Professor Brendan Mckay at the Shool of Computing in ANU developed a a practical algorithm for determining graph isomorphisms. He has implemented this algorithm as a software package.

- Practical graph isomorphism, II

The package is called **NAUTY**; stands for "**N**o **A**utomorphisms, **Y**es?"

# A research aside:

Emeritus Professor Brendan Mckay at the Shool of Computing in ANU developed a a practical algorithm for determining graph isomorphisms. He has implemented this algorithm as a software package.

- Practical graph isomorphism, II

The package is called **NAUTY**; stands for “**N**o **A**utomorphisms, **Y**es?”

It is commonly used by researchers for testing conjectures.

# Digraphs/Directed Graphs

# Digraphs

Digraphs were introduced in Section A3 in order to represent some relations diagrammatically. Here we look at digraphs in general.

# Digraphs

Digraphs were introduced in Section A3 in order to represent some relations diagrammatically. Here we look at digraphs in general.

A **directed graph** $G$ or **digraph** is a set $V(G)$ of **vertices** together with a multiset $E(G)$ with elements from $V(G) \times V(G)$ representing **directed edges**.

# Digraphs

Digraphs were introduced in Section A3 in order to represent some relations diagrammatically. Here we look at digraphs in general.

A **directed graph** $G$ or **digraph** is a set $V(G)$ of **vertices** together with a multiset $E(G)$ with elements from $V(G) \times V(G)$ representing **directed edges**.

- A **directed graph** (or **digraph**) is the same as a graph except that edges are *ordered* pairs of endpoints.

# Digraphs

Digraphs were introduced in Section A3 in order to represent some relations diagrammatically. Here we look at digraphs in general.

A **directed graph** $G$ or **digraph** is a set $V(G)$ of **vertices** together with a multiset $E(G)$ with elements from $V(G) \times V(G)$ representing **directed edges**.

- A **directed graph** (or **digraph**) is the same as a graph except that edges are *ordered* pairs of endpoints.
- Each edge has an **initial vertex** and a **final vertex**.

# Digraphs

Digraphs were introduced in Section A3 in order to represent some relations diagrammatically. Here we look at digraphs in general.

A **directed graph** $G$ or **digraph** is a set $V(G)$ of **vertices** together with a multiset $E(G)$ with elements from $V(G) \times V(G)$ representing **directed edges**.

- A **directed graph** (or **digraph**) is the same as a graph except that edges are *ordered* pairs of endpoints.
- Each edge has an **initial vertex** and a **final vertex**.
- Loops are still allowed.

# Digraphs

Digraphs were introduced in Section A3 in order to represent some relations diagrammatically. Here we look at digraphs in general.

A **directed graph** $G$ or **digraph** is a set $V(G)$ of **vertices** together with a multiset $E(G)$ with elements from $V(G) \times V(G)$ representing **directed edges**.

- A **directed graph** (or **digraph**) is the same as a graph except that edges are *ordered* pairs of endpoints.
- Each edge has an **initial vertex** and a **final vertex**.
- Loops are still allowed.
- The edges of a digraph are sometimes called **arcs**.

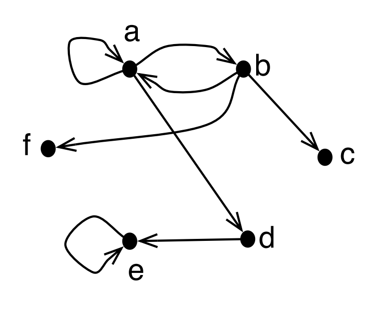
# Digraphs

Digraphs were introduced in Section A3 in order to represent some relations diagrammatically. Here we look at digraphs in general.

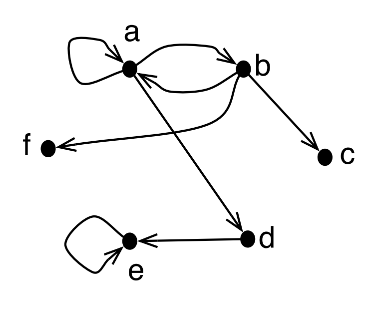
A **directed graph** $G$ or **digraph** is a set $V(G)$ of **vertices** together with a multiset $E(G)$ with elements from $V(G) \times V(G)$ representing **directed edges**.

- A **directed graph** (or **digraph**) is the same as a graph except that edges are *ordered* pairs of endpoints.
- Each edge has an **initial vertex** and a **final vertex**.
- Loops are still allowed.
- The edges of a digraph are sometimes called **arcs**.
- In a diagram of a digraph, the direction of an arc is given by an arrow.

# Example



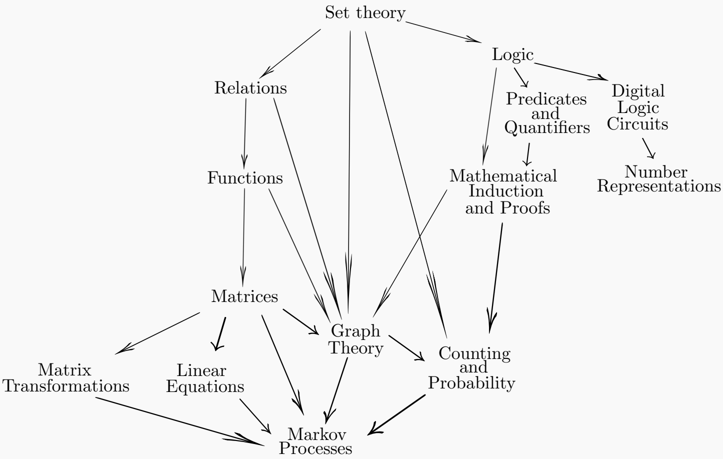
# Example




The vertex set and edge set for this graph are:

$$\begin{aligned} V(G) &= \{a, b, c, d, e, f\} \\ E(G) &= \{(a,a), (a,b), (a,d), (b,a), (b,c), (b,f), (d,e), (e,e)\} \end{aligned}$$

# An application: Recording Information Dependencies




An arrow from $A$ to $B$ means that $B$ depends upon $A$ in some way.

# Foodwebs

- An application of digraphs in ecology is in describing a *foodweb*.

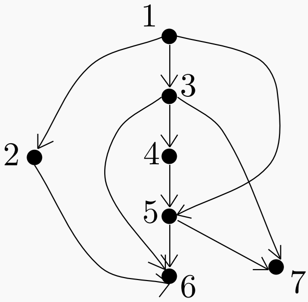
# Foodwebs

- An application of digraphs in ecology is in describing a *foodweb*.
- The next slide shows a foodweb developed by Parsons and LeBrasseur, as adapted by Cohen, pertaining to the following species in the Strait of Georgia, British Columbia.

| Key | Species |
|---|---|
| 1. | Juvenile pink salmon |
| 2. | P. Minutus |
| 3. | Calanusand Euphausiid Burcillia |
| 4. | Euphausiid Eggs |
| 5. | Euphausiids |
| 6. | Chaetoceros Socialis and Debilis |
| 7. | Mu-Flagellates |

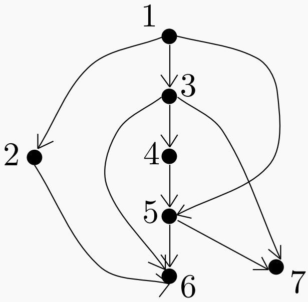
# Example

An arrow from $i$ to $j$ means *'$i$ eats $j$'* :



# Example

An arrow from $i$ to $j$ means *'$i$ eats $j$'*:

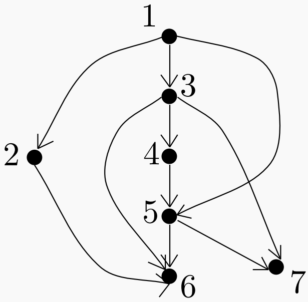



For example:

- species 1 eats species 2, 3, and 5

# Example

An arrow from $i$ to $j$ means *'$i$ eats $j$'* :

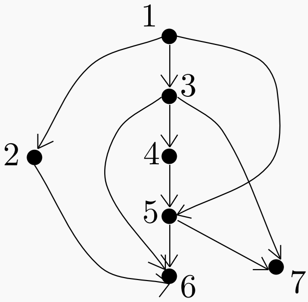



For example:

- species 1 eats species 2, 3, and 5
- species 4 *only* eats species 5.

# Example

An arrow from $i$ to $j$ means *'i eats j'* :



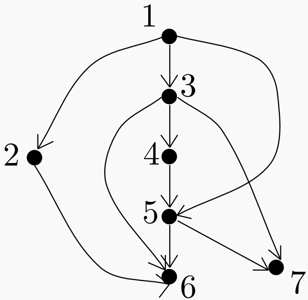

For example:

- species 1 eats species 2, 3, and 5
- species 4 *only* eats species 5.

Note: Some foodweb diagrams have their arrows *reversed*:
*i.e.* an arrow from $A$ to $B$ means '$A$ is food for $B$'.

# Example

An arrow from $i$ to $j$
means *'i eats j'* :




For example:

- species 1 eats species 2, 3, and 5
- species 4 *only* eats species 5.

Note: Some foodweb diagrams have their arrows *reversed*:
*i.e.* an arrow from $A$ to $B$ means '$A$ is food for $B$'.
We shall use the first convention unless stated otherwise.

# Niche Overlap Graphs

- An application of graphs in ecology is in describing commonalities (or competition) between species in a *Niche Overlap Graph*.

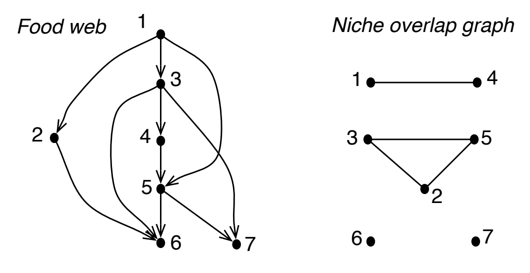
# Niche Overlap Graphs

- An application of graphs in ecology is in describing commonalities (or competition) between species in a *Niche Overlap Graph*.
- Each species is represented by a vertex. An undirected edge connects two vertices if and only if the species represented by these vertices compete for food.

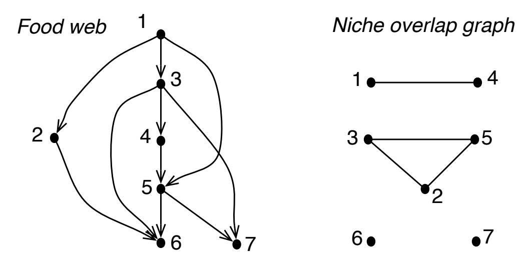
# Niche Overlap Graphs

- An application of graphs in ecology is in describing commonalities (or competition) between species in a *Niche Overlap Graph*.
- Each species is represented by a vertex. An undirected edge connects two vertices if and only if the species represented by these vertices compete for food.
- The following niche overlap graph is constructed from the Food Web data of the previous example.

# Food Webs and Niche Overlap Graphs



# Food Webs and Niche Overlap Graphs




**For example:**

Species 1 and 4 compete for food (species 5), so are connected by an edge in the niche overlap graph.

# Simple graphs

## Types of Graphs and Digraphs

Sometimes it is useful to restrict our attention to two types of graphs and digraphs, called:

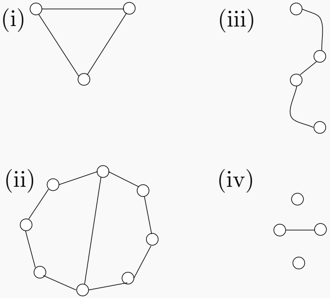
- **Simple Graphs** and
- **Simple Digraphs**

# Simple Graphs

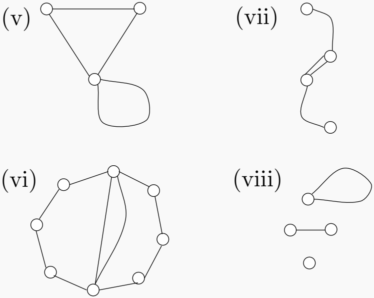
A **simple graph** is a graph that has no loops and no parallel edges.

# Simple Graphs

A **simple graph** is a graph that has no loops and no parallel edges.




*Some simple Graphs*




*Some non-simple Graphs*

# Terminology warning

- Warning: in graph theory, different authors use the *same words* to mean *different things*.

## Terminology warning

- Warning: in graph theory, different authors use the *same words* to mean *different things*.
- For some, what we are calling a simple graph is just a graph.

# Terminology warning

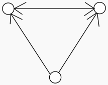
- Warning: in graph theory, different authors use the *same words* to mean *different things*.
- For some, what we are calling a simple graph is just a graph.
- For some, what we would call a graph with parallel edges, is a multi-graph.

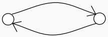
# Simple Digraphs

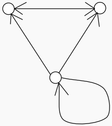
Similarly, a **simple digraph** is a digraph that has no loops and no parallel edges.

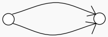
# Simple Digraphs

Similarly, a **simple digraph** is a digraph that has no loops and no parallel edges.

(i) 

(ii) 

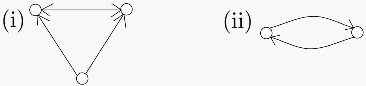
(iii) 

(iv) 

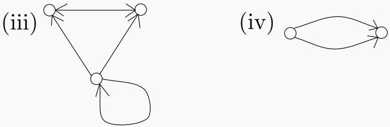
*Some simple Digraphs*

*Some non-simple Digraphs*

Similarly, a **simple digraph** is a digraph that has no loops and no parallel edges.




*Some simple Digraphs*



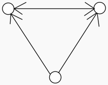

*Some non-simple Digraphs*

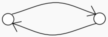
Note that:

- It is okay to have both $(a, b)$ and $(b, a)$ - these are not parallel;

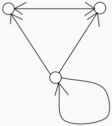
# Simple Digraphs

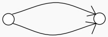
Similarly, a **simple digraph** is a digraph that has no loops and no parallel edges.

(i) 

(ii) 

*Some simple Digraphs*

(iii) 

(iv) 

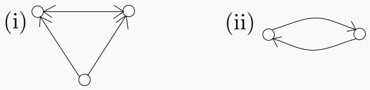
*Some non-simple Digraphs*

Note that:

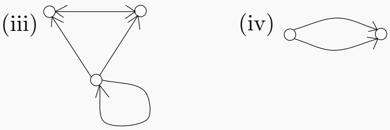
- It is okay to have both $(a, b)$ and $(b, a)$ - these are not parallel;
- It is not okay to have $(a, b)$ twice - those would be parallel.

# Simple Digraphs

Similarly, a **simple digraph** is a digraph that has no loops and no parallel edges.




*Some simple Digraphs*




*Some non-simple Digraphs*

Note that:

- It is okay to have both $(a, b)$ and $(b, a)$ - these are not parallel;
- It is not okay to have $(a, b)$ twice - those would be parallel.
- We sometimes draw a single edge with an arrow at each end to indicate a pair of edges $(a, b)$ and $(b, a)$, instead of drawing two distinct lines.