We will cover these parts of the book (8th edition):

9.4.1-9.4.4
9.5
9.6.1
10.1
10.2.1-10.2.4
10.2.7

Closures of Relations

- What is the **closure** of a relation?

- **Definition:** Let R be a relation on a set A. R may or may not have some **property P**, such as reflexivity, symmetry, or transitivity.

- If there is a relation S that contains R and has property P, and S is a subset of **every** subset of $A \times A$ relation that contains R and has property P, then S is called the **closure** of R with respect to P.

- **Note that the closure of a relation with respect to a property may not exist.**

Closures of Relations

- **Example I:** Find the **reflexive closure** of relation R = {(1, 1), (1, 2), (2, 1), (3, 2)} on the set A = {1, 2, 3}.

- **Solution:** We know that any reflexive relation on A must contain the elements (1, 1), (2, 2), and (3, 3).
- By adding (2, 2) and (3, 3) to R, we obtain the reflexive relation S, which is given by
S = {(1, 1), (1, 2), (2, 1), (2, 2), (3, 2), (3, 3)}.

- S is reflexive, contains R, and is contained within every reflexive relation that contains R.

- Therefore, S is the **reflexive closure** of R.

UMASS BOSTON

Closures of Relations

- **Example II:** Find the **symmetric closure** of the relation $R = \{(a, b) \mid a > b\}$ on the set of positive integers.

- **Solution:** The symmetric closure of R is given by
- $R \cup R^{-1} = \{(a, b) \mid a > b\} \cup \{(b, a) \mid a > b\}$
- $\qquad = \{(a, b) \mid a \neq b\}$

Closures of Relations

- **Example III:** Find the **transitive closure** of the relation R = {(1, 3), (1, 4), (2, 1), (3, 2)} on the set A = {1, 2, 3, 4}.
- **Solution:** R would be transitive, if for all pairs (a, b) and (b, c) in R there were also a pair (a, c) in R.
- If we add the missing pairs (1, 2), (2, 3), (2, 4), and (3, 1), will R be transitive?
- **No**, because the extended relation R contains (3, 1) and (1, 4), but does not contain (3, 4), (1, 1), (2, 2), (3, 3).
- By adding new elements to R, we also add **new requirements** for its transitivity. We need to look at **paths in digraphs** to solve this problem.

Closures of Relations

- Imagine that we have a relation R that represents all **train connections** in the US.
- For example, if (Boston, Philadelphia) is in R, then there is a **direct** train connection from Boston to Philadelphia.
- If R contains (Boston, Philadelphia) and (Philadelphia, Washington), there is an **indirect** connection from Boston to Washington.
- Because there are indirect connections, it is not possible by just looking at R to determine which cities are connected by trains.
- The transitive closure of R contains exactly those pairs of cities that are connected, **either directly or indirectly**.

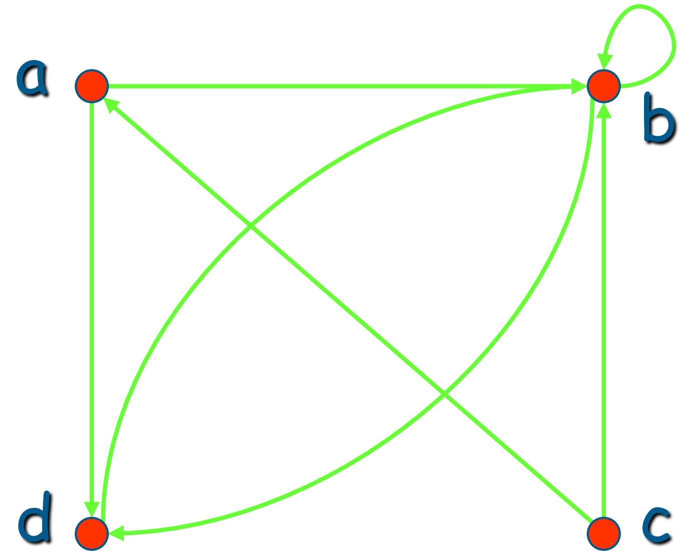
UMASS BOSTON

Closures of Relations

- **Definition:** A **path** from a to b in the directed graph G is a sequence of one or more edges (x_0, x_1), (x_1, x_2), (x_2, x_3), …, (x_{n-1}, x_n) in G, where $x_0 = a$ and $x_n = b$.
- In other words, a path is a **sequence of edges** where the terminal vertex of an edge is the same as the initial vertex of the next edge in the path.
- This path is denoted by $x_0, x_1, x_2, \ldots, x_n$ and has **length** n.
- A path that begins and ends at the same vertex is called a **circuit** or **cycle**.

Closures of Relations

- **Example:** Let us take a look at the following graph:



Is c,a,b,d,b a path in this graph? Yes

Is d,b,b,b,d,b,d a circuit in this graph? Yes

Is there any circuit including c in this graph? No

Closures of Relations

- Due to the one-to-one correspondence between graphs and relations, we can transfer the definition of path from graphs to relations:

- **Definition:** There is a path from a to b in a relation R, if there is a sequence of elements a, x_1, x_2, …, x_{n-1}, b with $(a, x_1) \in R$, $(x_1, x_2) \in R$, …, and $(x_{n-1}, b) \in R$.

- **Theorem:** Let R be a relation on a set A. There is a path from a to b if and only if $(a, b) \in R^n$ for some positive integer n.

Closures of Relations

- According to the train example, the transitive closure of a relation consists of the pairs of vertices in the associated directed graph that are connected by a path.

- **Definition:** Let R be a relation on a set A. The connectivity relation R* consists of the pairs (a, b) such that there is a path between a and b in R.

- We know that R^n consists of the pairs (a, b) such that a and b are connected by a path of length n.

- Therefore, R* is the union of R^n across all positive integers n:

$$R^* = \bigcup_{n=1}^{\infty} R^n = R^1 \cup R^2 \cup R^3 \cup \ldots$$

UMASS BOSTON

Closures of Relations

- **Theorem:** The transitive closure of a relation R equals the connectivity relation R*.

- But how can we compute R* ?

- **Lemma:** Let A be a set with n elements, and let R be a relation on A. If there is a path in R from a to b, then there is such a path with length not exceeding n.

- Moreover, if $a \neq b$ and there is a path in R from a to b, then there is such a path with length not exceeding $(n – 1)$.

Closures of Relations

- This lemma is based on the observation that if a path from a to b visits any vertex more than once, it must include at least one **circuit**.

- These circuits can be **eliminated** from the path, and the reduced path will still connect a and b.

- **Theorem:** For a relation R on a set A with n elements, the transitive closure R* is given by:
- $R^* = R \cup R^2 \cup R^3 \cup \ldots \cup R^n$
- For matrices representing relations we have:
- $M_{R^*} = M_R \vee M_R^{[2]} \vee M_R^{[3]} \vee \ldots \vee M_R^{[n]}$

UMASS BOSTON

Closures of Relations

- Let us finally solve **Example III** by finding the **transitive closure** of the relation R = {(1, 3), (1, 4), (2, 1), (3, 2)} on the set A = {1, 2, 3, 4}.

- R can be represented by the following matrix M_R:

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

Closures of Relations

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

$$M_R^{[2]} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$M_R^{[3]} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$M_R^{[4]} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$M_{R^*} = M_R \vee M_R^{[2]} \vee M_R^{[3]} \vee M_R^{[4]} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

UMASS BOSTON

Closures of Relations

- **Solution:** The transitive closure of the relation R = {(1, 3), (1, 4), (2, 1), (3, 2)} on the set A = {1, 2, 3, 4} is given by the relation

{(1, 1), (1, 2), (1, 3), (1, 4), (2, 1), (2, 2), (2, 3), (2, 4), (3, 1), (3, 2), (3, 3), (3, 4)}

Equivalence Relations

- **Equivalence relations** are used to relate objects that are similar in some way.

- **Definition:** A relation on a set A is called an equivalence relation if it is reflexive, symmetric, and transitive.

- Two elements that are related by an equivalence relation R are called **equivalent**.

- The notation $a \sim b$ is often used to denote that a and b are equivalent.

Equivalence Relations

- Since R is **symmetric**, a is equivalent to b whenever b is equivalent to a.
- Since R is **reflexive**, every element is equivalent to itself.
- Since R is **transitive**, if a and b are equivalent and b and c are equivalent, then a and c are equivalent.
- Obviously, these three properties are necessary for a reasonable definition of equivalence.

Equivalence Relations

- **Example:** Suppose that R is the relation on the set of strings that consist of English letters such that aRb if and only if l(a) = l(b), where l(x) is the length of the string x. Is R an equivalence relation?
- **Solution:**

• R is reflexive, because l(a) = l(a) and therefore aRa for any string a.

• R is symmetric, because if l(a) = l(b) then l(b) = l(a), so if aRb then bRa.

• R is transitive, because if l(a) = l(b) and l(b) = l(c), then l(a) = l(c), so aRb and bRc implies aRc.

- R is an equivalence relation.

Equivalence Classes

- **Definition:** Let R be an equivalence relation on a set A. The set of all elements that are related to an element a of A is called the **equivalence class** of a.

- The equivalence class of a with respect to R is denoted by $[a]_R$.

- When only one relation is under consideration, we will delete the subscript R and write **[a]** for this equivalence class.

- If $b \in [a]_R$, b is called a **representative** of this equivalence class.

Equivalence Classes

- **Example:** In the previous example (strings of identical length), what is the equivalence class of the word mouse, denoted by [mouse] ?

- **Solution:** [mouse] is the set of all English words containing five letters.

- For example, ‘horse’ would be a representative of this equivalence class.

Equivalence Classes

- **Theorem:** Let R be an equivalence relation on a set A. The following statements are equivalent:

(i) aRb

(ii) [a] = [b]

(iii) $[a] \cap [b] \neq \varnothing$

- **Reminder:** A **partition** of a set S is a collection of disjoint nonempty subsets of S that have S as their union. In other words, the collection of subsets A_i, $i \in I$, forms a partition of S if and only if

(i) $A_i \neq \varnothing$ for $i \in I$

(ii) $A_i \cap A_j = \varnothing$, if $i \neq j$

(iii) $\cup_{i \in I} A_i = S$

UMASS BOSTON

Equivalence Classes

- **Theorem:** Let R be an equivalence relation on a set S. Then the **equivalence classes** of R form a **partition** of S. Conversely, given a partition $\{A_i \mid i \in I\}$ of the set S, there is an equivalence relation R that has the sets A_i, $i \in I$, as its equivalence classes.

Equivalence Classes

- **Example:** Let us assume that Frank, Suzanne and George live in Boston, Stephanie and Max live in Lübeck, and Jennifer lives in Sydney.
- Let R be the **equivalence relation** {(a, b) | a and b live in the same city} on the set P = {Frank, Suzanne, George, Stephanie, Max, Jennifer}.
- Then R = {(Frank, Frank), (Frank, Suzanne), (Frank, George), (Suzanne, Frank), (Suzanne, Suzanne), (Suzanne, George), (George, Frank), (George, Suzanne), (George, George), (Stephanie, Stephanie), (Stephanie, Max), (Max, Stephanie), (Max, Max), (Jennifer, Jennifer)}.

Equivalence Classes

- Then the **equivalence classes** of R are:
- {{Frank, Suzanne, George}, {Stephanie, Max}, {Jennifer}}.
- This is a **partition** of P.

- The equivalence classes of any equivalence relation R defined on a set S constitute a partition of S, because every element in S is assigned to **exactly one** of the equivalence classes.

Equivalence Classes

- **Another example:** Let R be the relation $\{(a, b) \mid a \equiv b \pmod 3\}$ on the set of integers.
- Is R an equivalence relation?
- Yes, R is reflexive, symmetric, and transitive.
- What are the equivalence classes of R ?
- $\{\{\ldots, -6, -3, 0, 3, 6, \ldots\},$
  $\{\ldots, -5, -2, 1, 4, 7, \ldots\},$
  $\{\ldots, -4, -1, 2, 5, 8, \ldots\}\}$
- Again, these three classes form a partition of the set of integers.

Partial Orderings

- Sometimes, relations define an **order** on the elements in a set.
- **Definition:** A relation R on a set S is called a **partial ordering** or **partial order** if it is reflexive, antisymmetric, and transitive.
- A set S together with a partial ordering R is called a **partially ordered set**, or **poset**, and is denoted by (S, R). Members of S are called **elements** of the poset.

Partial Orderings

- **Example:** Consider the "greater than or equal" relation $\geq$ (defined by $\{(a, b) \mid a \geq b\}$).
- Is $\geq$ a **partial ordering** on the set of integers?
  - $\geq$ is **reflexive**, because $a \geq a$ for every integer a.
  - $\geq$ is **antisymmetric**, because if $a \neq b$, then $a \geq b \wedge b \geq a$ is false.
  - $\geq$ is **transitive**, because if $a \geq b$ and $b \geq c$, then $a \geq c$.
- Consequently, $(Z, \geq)$ is a partially ordered set.

UMASS BOSTON

Partial Orderings

- **Another example:** Is the "inclusion relation" $\subseteq$ a **partial ordering** on the power set of a set S?

- $\subseteq$ is **reflexive**, because $A \subseteq A$ for every set A.
- $\subseteq$ is **antisymmetric**, because if $A \neq B$, then $A \subseteq B \wedge B \subseteq A$ is false.
- $\subseteq$ is **transitive**, because if $A \subseteq B$ and $B \subseteq C$, then $A \subseteq C$.

- Consequently, $(P(S), \subseteq)$ is a partially ordered set.

Partial Orderings

- In a poset the notation $a \leq b$ denotes that $(a, b) \in R$.
- Note that the symbol $\leq$ is used to denote the relation in **any** poset, not just the "less than or equal" relation.
- The notation $a < b$ denotes that $a \leq b$, but $a \neq b$.
- If $a < b$ we say "a is less than b" or "b is greater than a".

Partial Orderings

- For two elements a and b of a poset $(S, \leq)$ it is possible that neither $a \leq b$ nor $b \leq a$.
- **Example:** In $(P(Z), \subseteq)$, $\{1, 2\}$ is not related to $\{1, 3\}$, and vice versa, since neither is contained within the other.
- **Definition:** The elements a and b of a poset $(S, \leq)$ are called **comparable** if either $a \leq b$ or $b \leq a$. When a and b are elements of S such that neither $a \leq b$ nor $b \leq a$, then a and b are called **incomparable**.

UMASS BOSTON

Partial Orderings

- For some applications, we require **all** elements of a set to be comparable.
- For example, if we want to write a dictionary, we need to define an order on **all** English words (alphabetic order).
- **Definition:** If (S, ≤) is a poset and **every two elements** of S are comparable, S is called a **totally ordered** or **linearly ordered set**, and ≤ is called a **total order** or **linear order**. A totally ordered set is also called a **chain**.

Partial Orderings

- **Example I:** Is (Z, $\leq$) a totally ordered set?
- Yes, because $a \leq b$ or $b \leq a$ for all integers a and b.

- **Example II:** Is (Z^+, $|$) a totally ordered set?
- No, because it contains incomparable elements such as 5 and 7.

Partial Orderings

- **Definition:** $(S, \leq)$ is a **well-ordered set** if it is a poset such that $\leq$ is a total ordering and every nonempty subset of S has a least element.

- **Example I:** The set of ordered pairs of positive integers ($Z^+ \times Z^+$), with $(a_1,a_2) \leq (b_1,b_2)$ if $a_1<b_1$, or if $a_1=b_1$ and $a_2 \leq b_2$ (the lexicographic ordering) is a well-ordered set. Because it is a total ordering and (1,1) is the least element.

Partial Orderings

- **The Principle of Well-ordered Induction:**
- Suppose that S is a well-ordered set. Then P(x) is true for all $x \in S$, if
  - Inductive step: For every $y \in S$, if P(x) is true for all $x \in S$ with $x < y$, then P(y) is true.

- **Proof:** Suppose it is not the case that P(x) is true for all $x \in S$. Then there is an element $y \in S$ such that P(y) is false. Consequently, the set $A = \{x \in S \mid P(x) \text{ is false}\}$ is nonempty.

Partial Orderings

- Because S is well-ordered, A has a least element a. By the choice of a as a least element of A, we know that $P(x)$ is true for all $x \in S$ with $x < a$. This implies by the inductive step $P(a)$ is true. This contradiction shows that $P(x)$ must be true for all $x \in S$.

Let us switch to a new topic:

- **Graphs**

Introduction to Graphs

- **Definition:** A **simple graph** G = (V, E) consists of V, a nonempty set of vertices (or nodes), and E, a set of **unordered pairs** of distinct elements of V called edges. Each edge has either one or two vertices associated with it, called its endpoints. An edge is said to connect its endpoints.

- A simple graph is just like a directed graph, but with no specified direction of its edges.

- Sometimes we want to model **multiple connections** between vertices, which is impossible using simple graphs.

- In these cases, we have to use **multigraphs**.

Introduction to Graphs

- **Definition:** A **multigraph** $G = (V, E)$ consists of a set V of vertices, a set E of edges, and a function f from E to $\{\{u, v\} \mid u, v \in V, u \neq v\}$.
- The edges e_1 and e_2 are called **multiple or parallel edges** if $f(e_1) = f(e_2)$.
- **Note:**

• Edges in multigraphs are not necessarily defined as pairs, but can be of any type.

• No loops are allowed in multigraphs ($u \neq v$).

Introduction to Graphs

- **Example:** A multigraph G with vertices V = {a, b, c, d}, edges {1, 2, 3, 4, 5} and function f with f(1) = {a, b}, f(2) = {a, b}, f(3) = {b, c}, f(4) = {c, d} and f(5) = {c, d}:

Introduction to Graphs

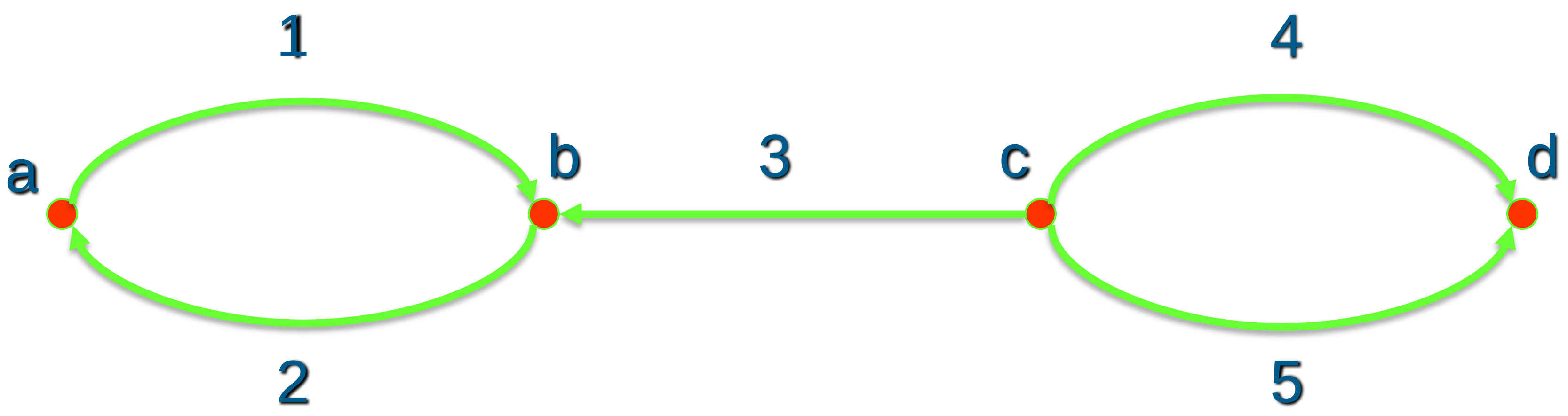
- If we want to define loops, we need the following type of graph:
- **Definition:** A **pseudograph** $G = (V, E)$ consists of a set V of vertices, a set E of edges, and a function f from E to $\{\{u, v\} \mid u, v \in V\}$.
- An edge e is a loop if $f(e) = \{u, u\}$ for some $u \in V$.

Introduction to Graphs

- Here is a type of graph that we already know:
- **Definition:** A **directed graph** $G = (V, E)$ consists of a set V of vertices and a set E of edges that are ordered pairs of elements in V.
- … leading to a new type of graph:
- **Definition:** A **directed multigraph** $G = (V, E)$ consists of a set V of vertices, a set E of edges, and a function f from E to $\{(u, v) \mid u, v \in V\}$.
- The edges e_1 and e_2 are called **multiple edges** if $f(e_1) = f(e_2)$.

Introduction to Graphs

- **Example:** A directed multigraph G with vertices V = {a, b, c, d}, edges {1, 2, 3, 4, 5} and function f with f(1) = (a, b), f(2) = (b, a), f(3) = (c, b), f(4) = (c, d) and f(5) = (c, d):



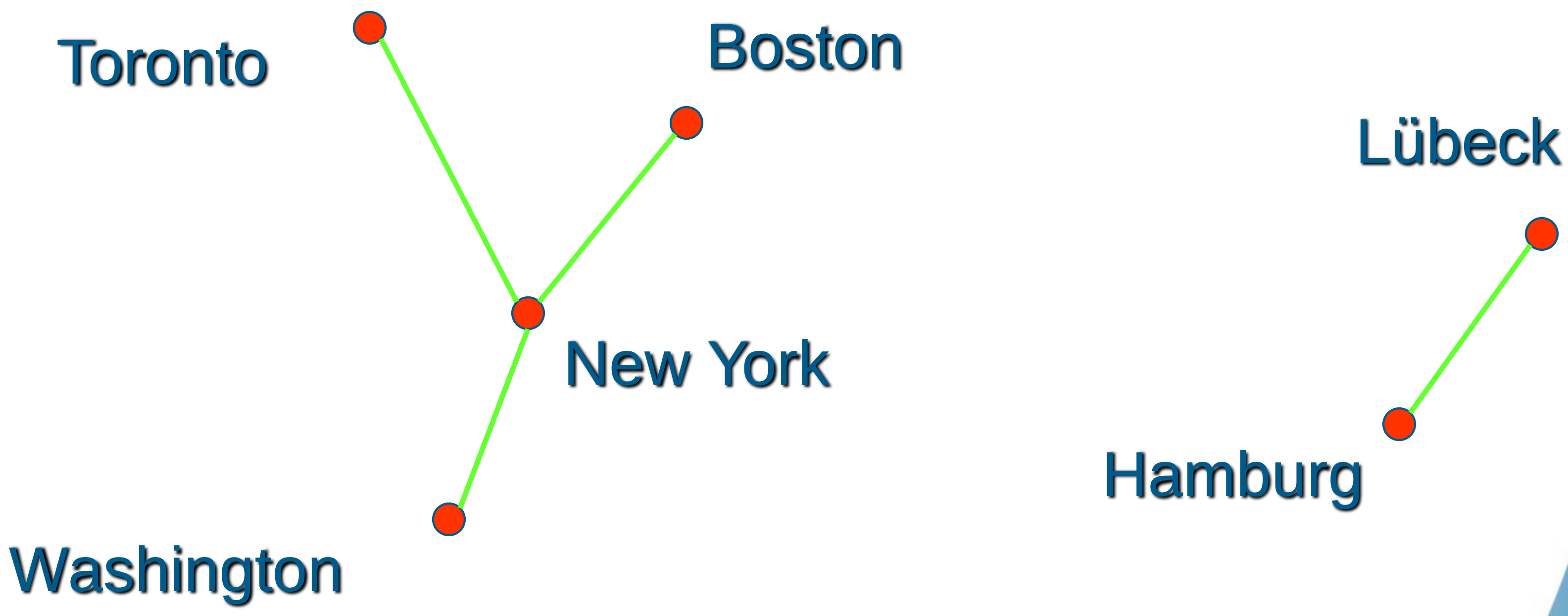
Introduction to Graphs

- Types of Graphs and Their Properties

Type	Edges	Multiple Edges?	Loops?
simple graph	undirected	no	no
multigraph	undirected	yes	no
pseudograph	undirected	yes	yes
simple dir. graph	directed	no	no
dir. multigraph	directed	yes	yes
mixed graph	dir and undir	yes	yes

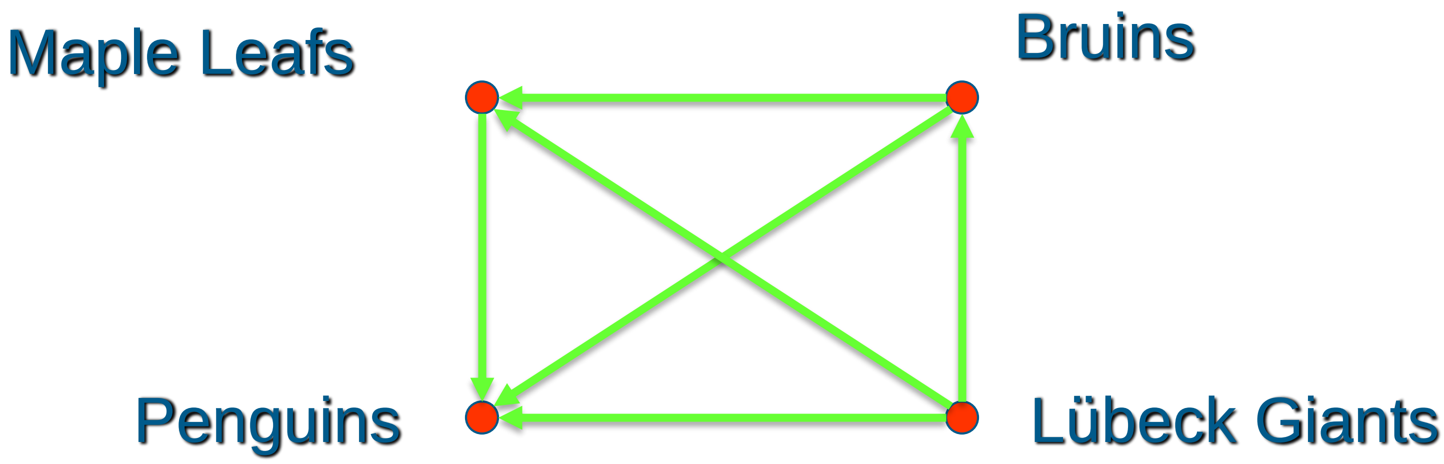
Graph Models

- **Example I:** How can we represent a network of (bi-directional) railways connecting a set of cities?
- We should use a **simple graph** with an edge {a, b} indicating a direct train connection between cities a and b.



Graph Models

- **Example II:** In a round-robin tournament, each team plays against each other team exactly once. How can we represent the results of the tournament (which team beats which other team)?
- We should use a **directed graph** with an edge (a, b) indicating that team a beats team b.



Graph Terminology

- **Definition:** Two vertices u and v in an undirected graph G are called **adjacent** (or **neighbors**) in G if {u, v} is an edge in G.

- If e = {u, v}, the edge e is called **incident with** the vertices u and v. The edge e is also said to **connect** u and v.

- The vertices u and v are called **endpoints** of the edge {u, v}.

Graph Terminology

- **Definition:** The set of all neighbors of a vertex v of G = (V, E), denoted by N(v), is called the **neighborhood** of v.

- If A is a subset of V, we denote by N(A) the set of all vertices in G that are adjacent to at least one vertex in A. So, $N(A) = \bigcup_{v \in A} N(v)$.

Graph Terminology

- **Definition:** The **degree** of a vertex in an undirected graph is the number of edges incident with it, except that a loop at a vertex contributes twice to the degree of that vertex.

- In other words, you can determine the degree of a vertex in a displayed graph by **counting the lines** that touch it.

- The degree of the vertex v is denoted by **deg(v).**

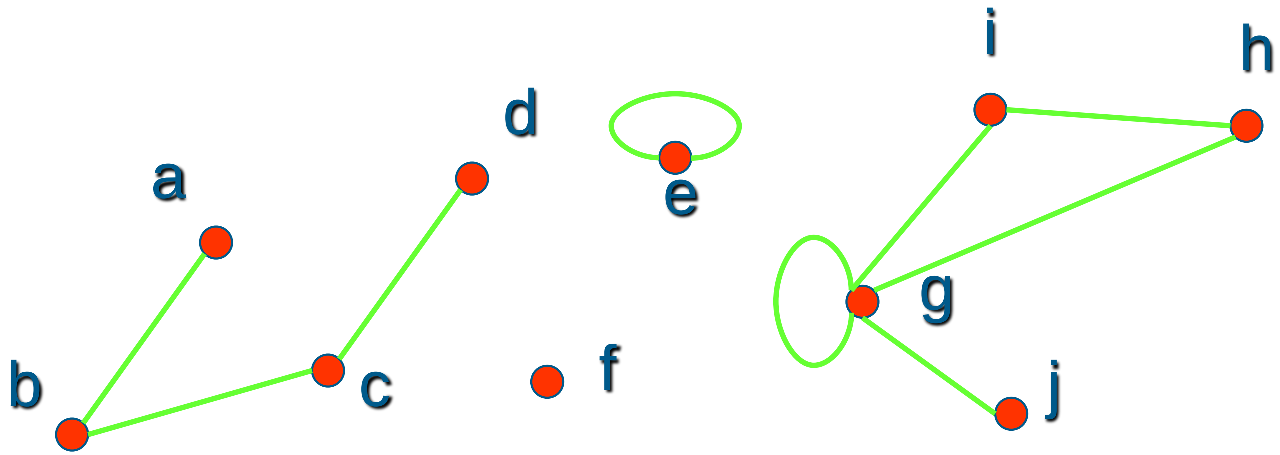
UMASS BOSTON

Graph Terminology

- A vertex of degree 0 is called **isolated**, since it is not adjacent to any vertex.
- **Note:** A vertex with a **loop** at it has at least degree 2 and, by definition, is **not isolated**, even if it is not adjacent to any **other** vertex.
- A vertex of degree 1 is called **pendant**. It is adjacent to exactly one other vertex.

Graph Terminology

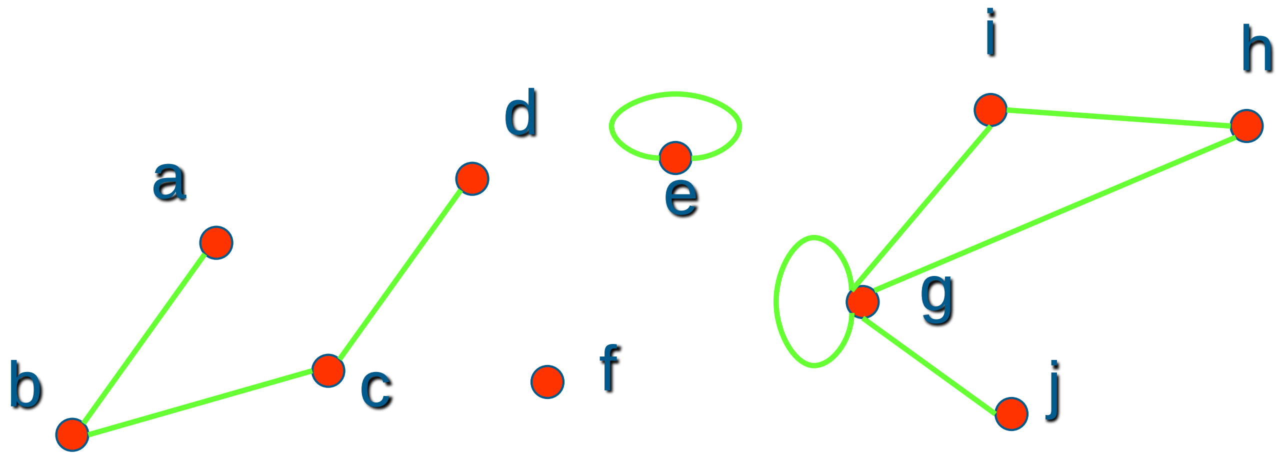
- **Example:** Which vertices in the following graph are isolated, which are pendant, and what is the maximum degree? What type of graph is it?



Solution: Vertex f is isolated, and vertices a, d and j are pendant. The maximum degree is deg(g) = 5.
This graph is a pseudograph (undirected, loops).

Graph Terminology

- Let us look at the same graph again and determine the number of its edges and the sum of the degrees of all its vertices:



Result: There are 9 edges, and the sum of all degrees is 18. This is easy to explain: Each new edge increases the sum of degrees by exactly two.

Graph Terminology

- **The Handshaking Theorem:** Let G = (V, E) be an undirected graph with e edges. Then
- $2e = \sum_{v \in V} \deg(v)$
- **Note:** This theorem holds even if multiple edges and/or loops are present.

- **Example:** How many edges are there in a graph with 10 vertices, each of degree 6?
- **Solution:** The sum of the degrees of the vertices is $6 \cdot 10 = 60$. According to the Handshaking Theorem, it follows that $2e = 60$, so there are 30 edges.

UMASS BOSTON

Graph Theorems

- **Theorem:** An undirected graph has an even number of vertices of odd degree.
- **Idea:** There are three possibilities for adding an edge to connect two vertices in the graph:
- **Before:** **After:**

Before		After
Both vertices have even degree	⇨	Both vertices have odd degree
Both vertices have odd degree	⇨	Both vertices have even degree
One vertex has odd degree, the other even	⇨	One vertex has even degree, the other odd

Graph Theorems

- There are two possibilities for adding a loop to a vertex in the graph:

Before:		**After:**
The vertex has even degree	$\Rightarrow$	The vertex has even degree
The vertex has odd degree	$\Rightarrow$	The vertex has odd degree

Graph Terminology

- So if there is an even number of vertices of odd degree in the graph, it will still be even after adding an edge.

- Therefore, since an undirected graph with **no edges** has an even number of vertices with odd degree (zero), the same must be true for **any** undirected graph.

Graph Terminology

- **Definition:** When (u, v) is an edge of the graph G with directed edges, u is said to be **adjacent to** v, and v is said to be **adjacent from** u.

- The vertex u is called the **initial vertex** of (u, v), and v is called the **terminal vertex** of (u, v).

- The initial vertex and terminal vertex of a loop are the same.

Graph Terminology

- **Definition:** In a graph with directed edges, the **in-degree** of a vertex v, denoted by **$\deg^-(v)$**, is the number of edges with v as their **terminal vertex**.

- The **out-degree** of v, denoted by **$\deg^+(v)$**, is the number of edges with v as their initial vertex.

- **Question:** How does adding a loop to a vertex change the in-degree and out-degree of that vertex?

- **Answer:** It increases both the in-degree and the out-degree by one.

UMASS BOSTON

Graph Terminology

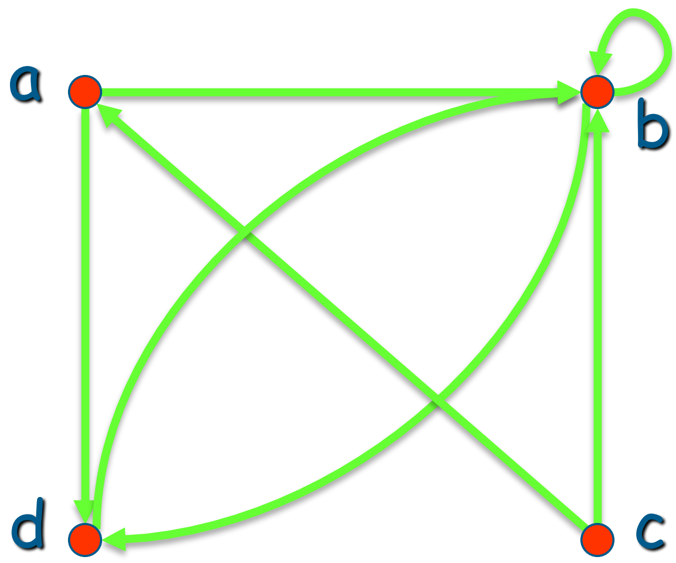
- **Example:** What are the in-degrees and out-degrees of the vertices a, b, c, d in this graph:

$\deg^-(a) = 1$

$\deg^+(a) = 2$

$\deg^-(b) = 4$

$\deg^+(b) = 2$



$\deg^-(d) = 2$

$\deg^+(d) = 1$

$\deg^-(c) = 0$

$\deg^+(c) = 2$

Graph Terminology

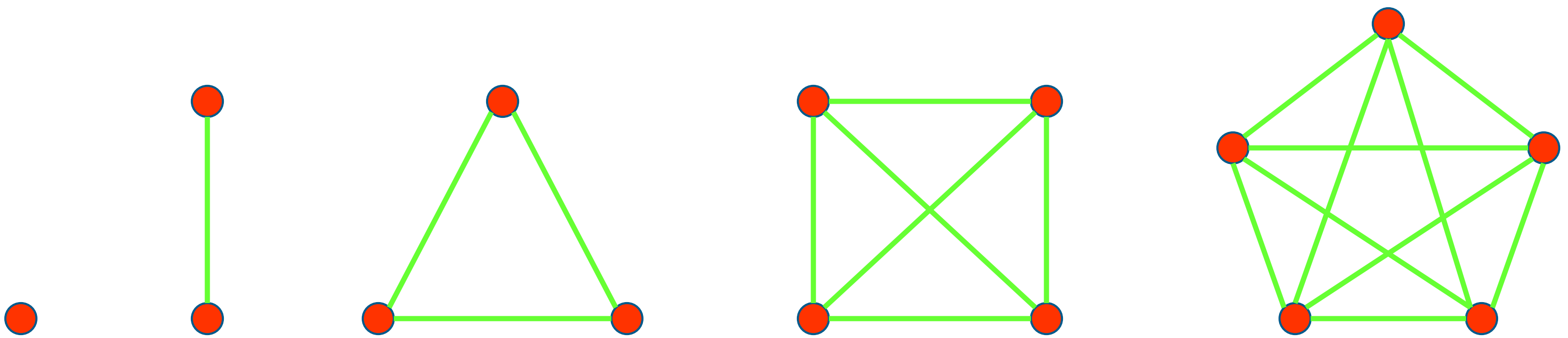
- **Theorem:** Let G = (V, E) be a graph with directed edges. Then:
- $\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = |E|$

- This is easy to see, because every new edge increases both the sum of in-degrees and the sum of out-degrees by one.

UMASS BOSTON

Special Graphs

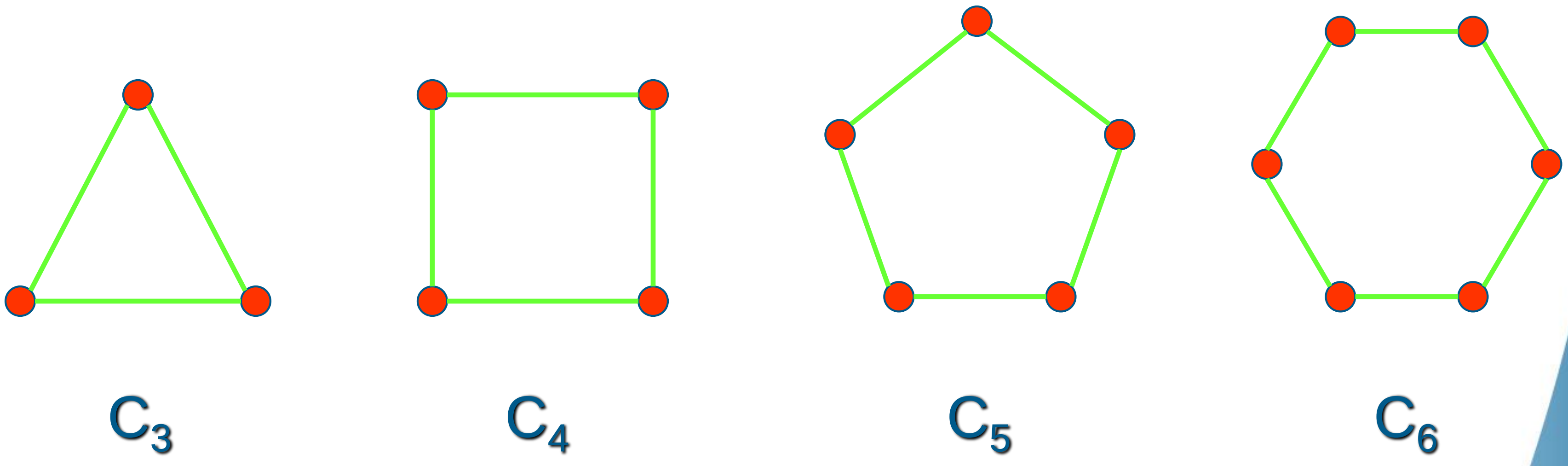
- **Definition:** The **complete graph** on n vertices, denoted by K_n, is the simple graph that contains exactly one edge between each pair of distinct vertices.



K_1 K_2 K_3 K_4 K_5

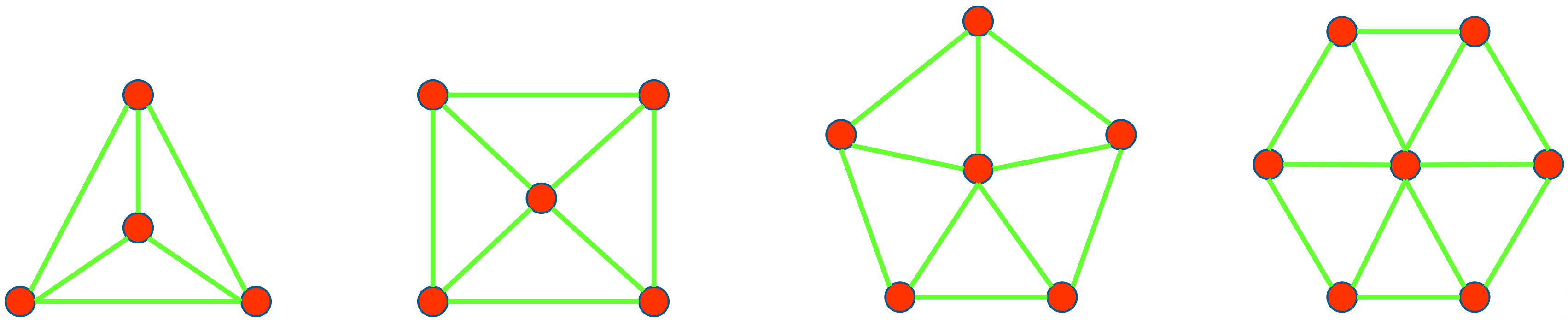
Special Graphs

- **Definition:** The **cycle** C_n, $n \geq 3$, consists of n vertices $v_1, v_2, \ldots, v_n$ and edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$.



Special Graphs

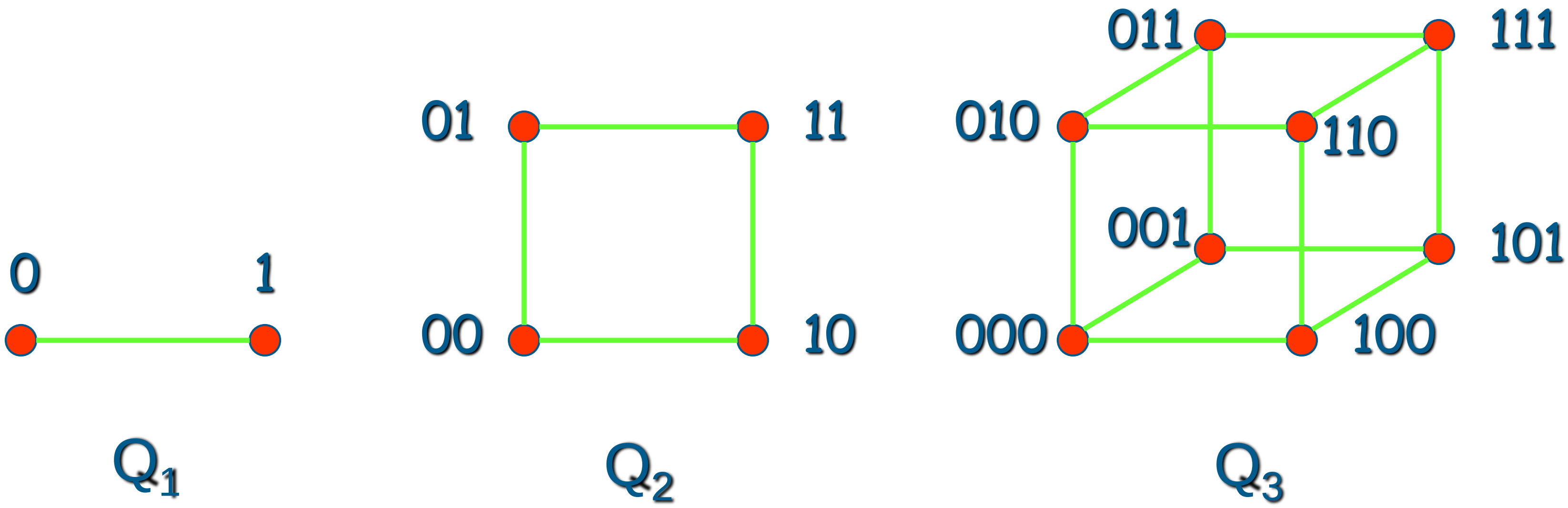
- **Definition:** We obtain the **wheel** W_n when we add an additional vertex to the cycle C_n, for $n \geq 3$, and connect this new vertex to each of the n vertices in C_n by adding new edges.



W_3 W_4 W_5 W_6

Special Graphs

- **Definition:** The **n-cube,** denoted by Q_n, is the graph that has vertices representing the 2^n bit strings of length n. Two vertices are adjacent if and only if the bit strings that they represent differ in exactly one bit position.



Special Graphs

- **Definition:** A simple graph is called **bipartite** if its vertex set V can be partitioned into two disjoint nonempty sets V_1 and V_2 such that every edge in the graph connects a vertex in V_1 with a vertex in V_2 (so that no edge in G connects either two vertices in V_1 or two vertices in V_2).

- When this condition holds, we call the pair (V1, V2) a **bipartition** of the vertex set V of G.

UMASS BOSTON

Special Graphs

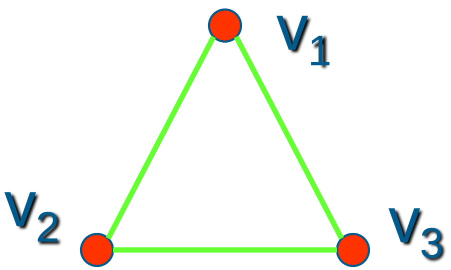
- For example, consider a graph that represents each person in a mixed-doubles tennis tournament (i.e., teams consist of one female and one male player).

- Players of the same team are connected by edges.

- This graph is **bipartite**, because each edge connects a vertex in the **subset of males** with a vertex in the **subset of females**.

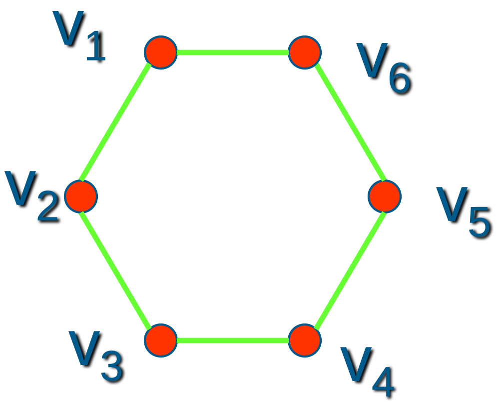
Special Graphs

- **Example I:** Is C_3 bipartite?

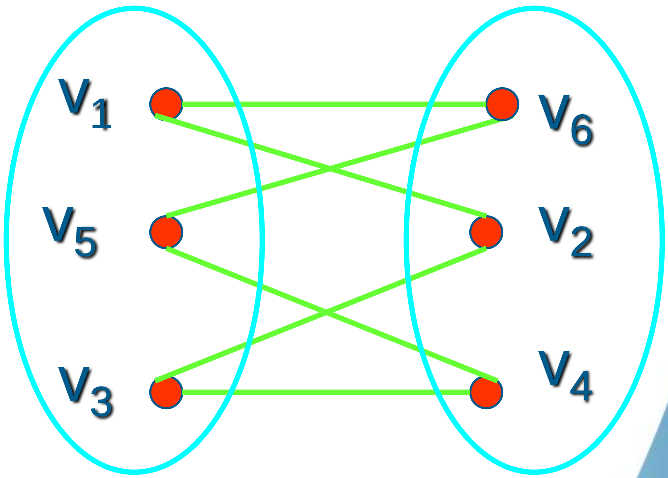


No, because there is no way to partition the vertices into two sets so that there are no edges with both endpoints in the same set.

Example II: Is C_6 bipartite?

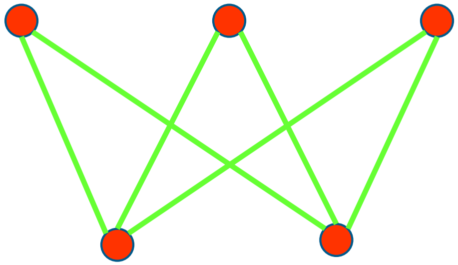


Yes, because we can display C_6 like this:

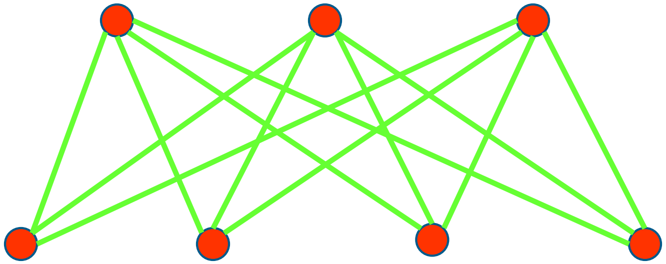


Special Graphs

- **Definition:** The **complete bipartite graph** $K_{m,n}$ is the graph that has its vertex set partitioned into two subsets of m and n vertices, respectively. Two vertices are connected if and only if they are in different subsets.



$K_{3,2}$



$K_{3,4}$

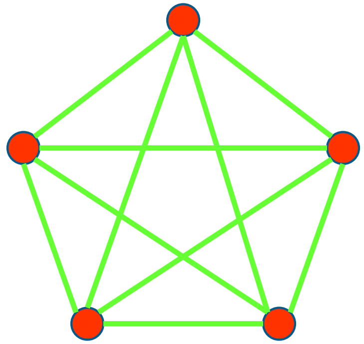
Operations on Graphs

- **Definition:** A **subgraph** of a graph G = (V, E) is a graph H = (W, F) where W⊆V and F⊆E.

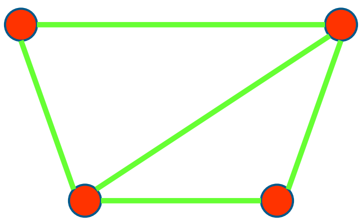
- A subgraph H of G is a proper subgraph of G if H≠G.

- **Note:** Of course, H is a valid graph, so we cannot remove any endpoints of remaining edges when creating H.

- **Example:**



K_5

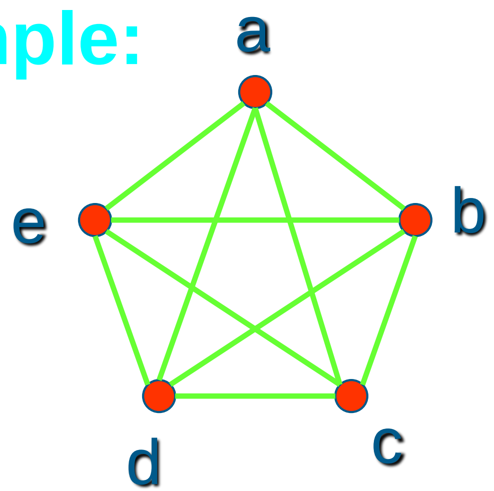


subgraph of K_5

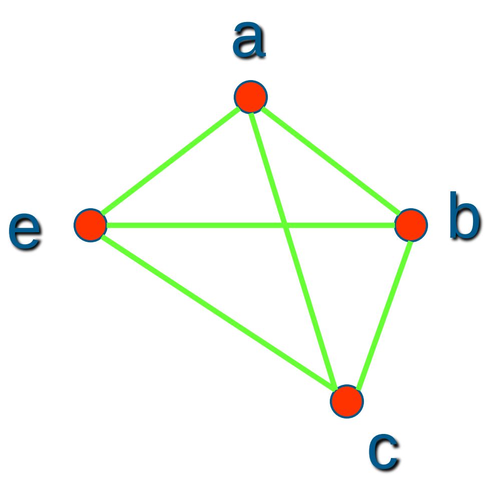
Operations on Graphs

- **Definition:** The **subgraph induced** by a subset W of the vertex set V is the graph (W,F), where the edge set F contains an edge in E if and only if both endpoints of this edge are in W.

- **Example:**



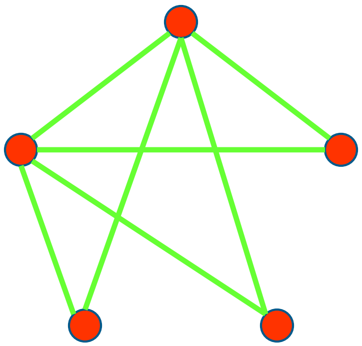
K_5



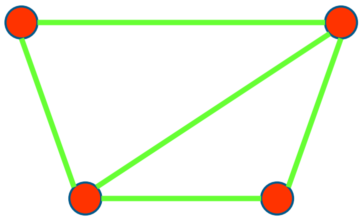
Subgraph induced by W={a,b,c,e} of K_5

Operations on Graphs

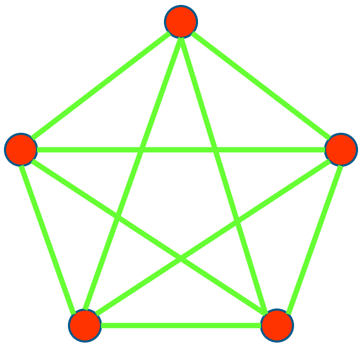
- **Definition:** The **union** of two simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the simple graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$.
- The union of G_1 and G_2 is denoted by $\mathbf{G_1 \cup G_2}$.



G_1



G_2



$G_1 \cup G_2 = K_5$