Relations

Chapter 9

Mr. SHOAIB RAZA

Chapter Summary

- Relations and Their Properties
- Representing Relations
- Equivalence Relations
- Partial Orderings

Relations and Their Properties

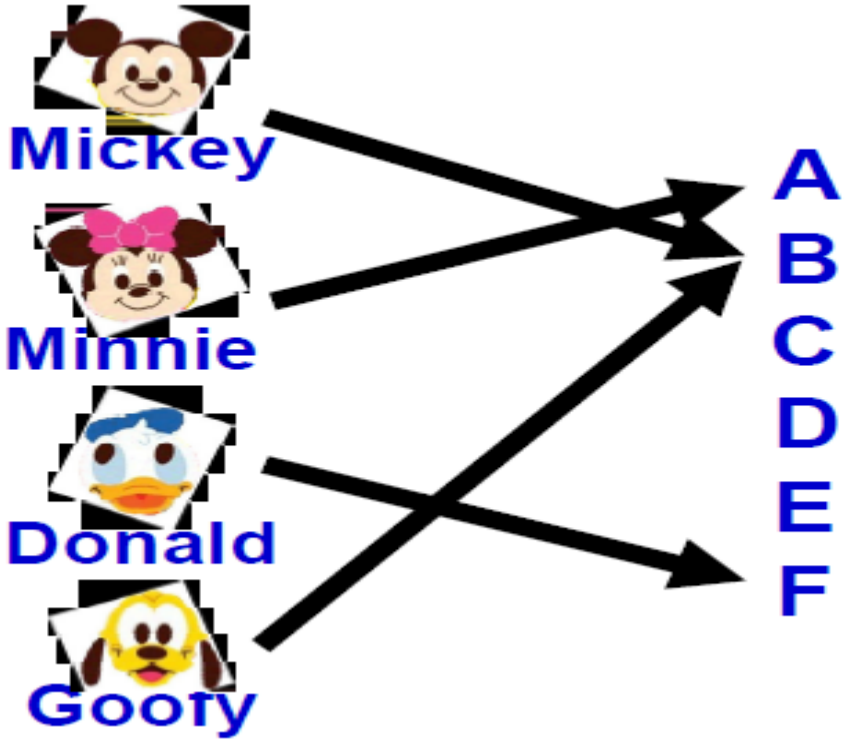
Section 9.1

Section Summary

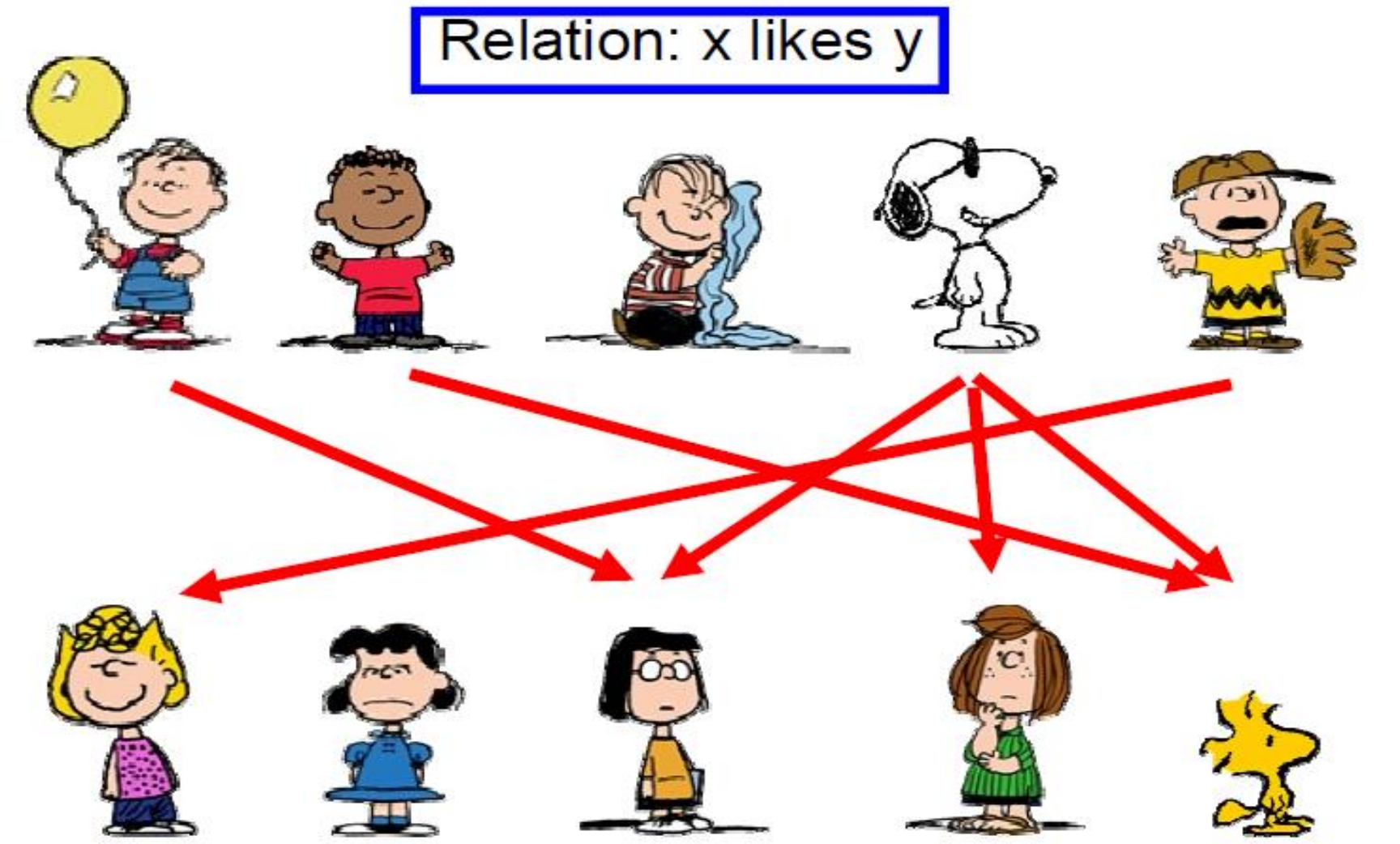
- Relations and Functions
- Properties of Relations
  - Reflexive Relations
  - Symmetric Relations
  - Antisymmetric Relations
  - Transitive Relations
  - Irreflexive Relations
  - Asymmetric Relations
- Combining Relations

Recall, Function is…

- Let **A** and **B** be nonempty sets Function ***f*** from **A** to **B** is an assignment of exactly one element of **B** to each element of **A.**
- By **defining** using a **relation**, a **function** from A to B contains **unique** ordered pair (a, b) for **every** element a ∈ A.



What is Relation?



Binary Relations

Definition: A *binary relation R* from a set *A* to a set *B* is a subset $R \subseteq A \times B$.

- Recall, for example:

  A = {a1, a2} and B = {b1, b2, b3}

  A x B = { (a1, b1), (a1, b2), (a1, b3),
  (a2, b2), (a1, b2), (a1, b3)}

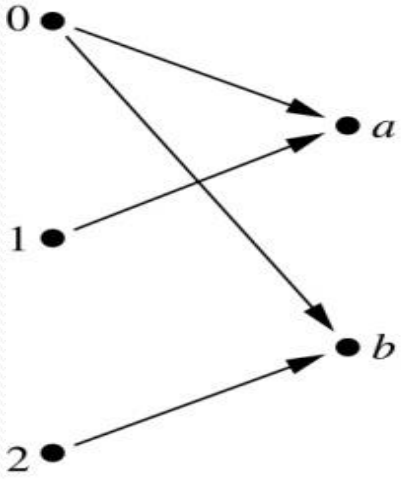
- Ordered pairs, which
  - First element comes from A
  - Second element comes from B
  - **aRb**: (a, b) ∈ R
  - **aRb** : (a, b) ∉ R

Moreover, when (a, b) belongs to R, a is said to be related to b by R.

Binary Relations

Example:

- Let $A = \{0,1,2\}$ and $B = \{a,b\}$
- $\{(0, a), (0, b), (1,a) , (2, b)\}$ is a relation from A to B.
- We can represent relations from a set A to a set B graphically or using a table:



R	a	b
0	×	×
1	×	
2		×

Binary Relations

EXAMPLE:

- Let A = {eggs, milk, corn} and B = {cows, goats, hens}

Define a relation R from A to B by (a, b) $\in$R *iff* a is produced by b.

- Then R = {(eggs, hens), (milk, cows), (milk, goats)}
- Thus, with respect to this relation eggs R hens , milk R cows, etc.

Binary Relations

EXAMPLE #1:

- S = {Peter, Paul, Mary}
- C = {C++, DisMath}

- Given
  - Peter takes C++ — Peter R C++ — Peter $\not R$ DisMath
  - Paul takes DisMath — Paul $\not R$ C++ — Paul R DisMath
  - Mary takes none of them — Mary $\not R$ C++ — Mary $\not R$ DisMath

- R = {(Peter, C++), (Paul, DisMath)}

Domain and Range of a Relation

DOMAIN OF A RELATION:

The domain of a relation R from A to B is the set of all first elements of the ordered pairs which belong to R denoted by Dom(R).

Symbolically, $Dom(R) = \{a \in A \mid (a, b) \in R\}$

RANGE OF A RELATION:

The range of a relation R from A to B is the set of all second elements of the ordered pairs which belong to R denoted Ran(R).

Symbolically, $Ran(R) = \{b \in B \mid (a, b) \in R\}$

Domain and Range of a Relation

EXERCISE:

Let A = {1, 2}, B = {1, 2, 3},

Define a binary relation R from A to B as follows:

$R = \{(a, b) \in A \times B \mid a < b\}$ Then

- a. Find the ordered pairs in R.
- b. Find the Domain and Range of R.
- c. Is 1R3, 2R2?

SOLUTION:

Given A = {1, 2}, B = {1, 2, 3},

$A \times B = \{(1,1), (1,2), (1,3), (2,1), (2,2), (2,3)\}$

- a. $R = \{(a, b) \in A \times B \mid a < b\}$

  $R = \{(1,2), (1,3), (2,3)\}$

Domain and Range of a Relation

Given A = {1, 2}, B = {1, 2, 3},

$$A \times B = \{(1,1), (1,2), (1,3), (2,1), (2,2), (2,3)\}$$

- b. Find the Domain and Range of R.

Solution:

Dom(R) = {1,2} and Ran(R) = {2, 3}

- c. Is 1R3, 2R2?

Solution:

c. Since (1, 3)∈R so 1R3.

But (2, 2) ∉R so 2 is not related with3 or $2 \not R\, 2$

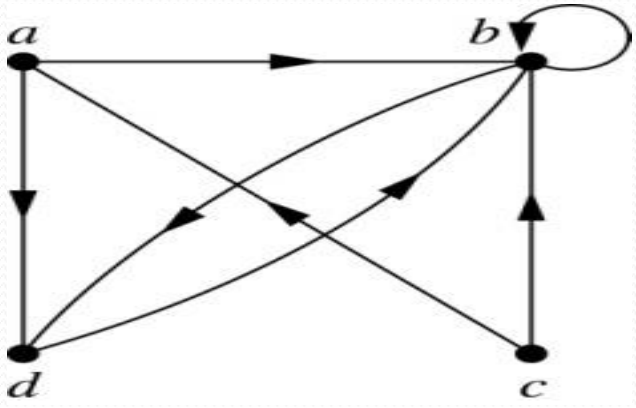
Representing Relations

Representing Relations Using Digraphs

Definition: A *directed graph*, or *digraph*, consists of a set V of *vertices* (or *nodes*) together with a set E of ordered pairs of elements of V called *edges* (or *arcs*). The vertex a is called the *initial vertex* of the edge (a,b), and the vertex b is called the *terminal vertex* of this edge.

- An edge of the form (a,a) is called a *loop*.

Example: A drawing of the directed graph with vertices a, b, c, and d, and edges (a, b), (a, d), (b, b), (b, d), $(c,$ a$)$, (c, b), and (d, b) is shown here.



Representing Relations Using Matrices

- A relation between finite sets can be represented using a zero-one matrix.
- Suppose R is a relation from $A = \{a_1, a_2, \ldots, a_m\}$ to $B = \{b_1, b_2, \ldots, b_n\}$.
  - The elements of the two sets can be listed in any particular arbitrary order. When $A = B$, we use the same ordering.
- The relation R is represented by the matrix $M_R = [m_{ij}]$, where

$$m_{ij} = \begin{cases} 1 \text{ if } (a_i, b_j) \in R, \\ 0 \text{ if } (a_i, b_j) \notin R. \end{cases}$$

- The matrix representing R has a 1 as its (i,j) entry when a_i is related to b_j and a 0 if a_i is not related to b_j.

Examples of Representing Relations Using Matrices

Example 1: Suppose that $A = \{1,2,3\}$ and $B = \{1,2\}$. Let R be the relation from A to B containing (a,b) if $a \in A$, $b \in B$, and $a > b$. What is the matrix representing R (assuming the ordering of elements is the same as the increasing numerical order)?

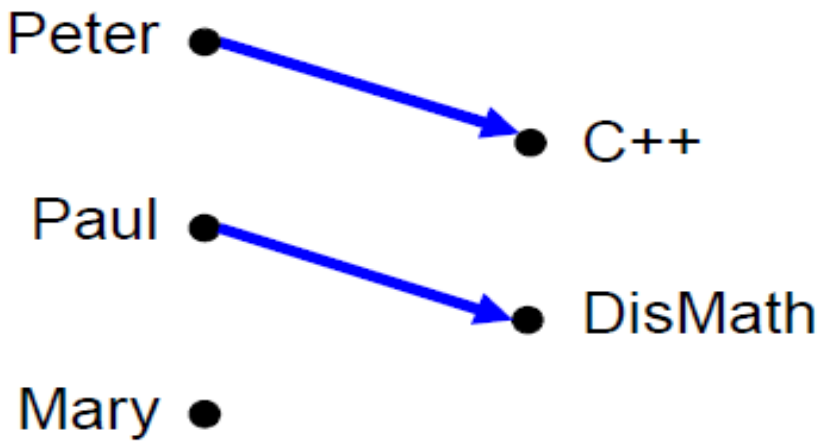
Solution: Because $R = \{(2,1), (3,1),(3,2)\}$, the matrix is

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

Binary Relations

EXAMPLE #1: (cont.)

- Peter R C++, Peter $\not R$ DisMath
  Paul $\not R$ C++, Paul R DisMath
  Mary $\not R$ C++, Mary $\not R$ DisMath



Directed Graph

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

Matrix

Binary Relation on a Set

Definition: A binary relation *R on a set A* is a subset of $A \times A$ or a relation from A to A.

Example:

- Suppose that $A = \{a,b,c\}$. Then $R = \{(a,a),(a,b),(a,c)\}$ is a relation on A.
- Let $A = \{1, 2, 3, 4\}$. The ordered pairs in the relation $R = \{(a,b) \mid a \text{ divides } b\}$ are

  $(1,1), (1, 2), (1,3), (1, 4), (2, 2), (2, 4), (3, 3)$, and $(4, 4)$.

REMARK:

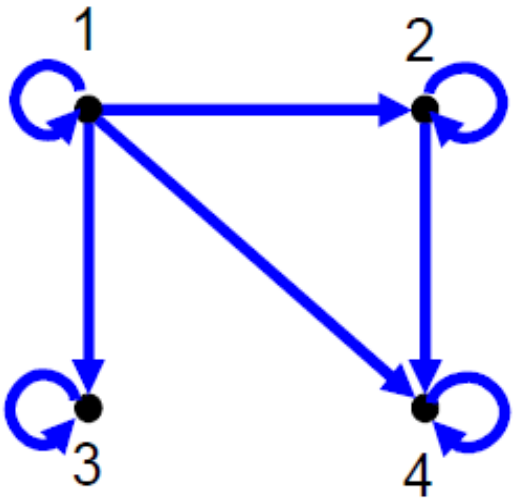
For any set A

1. A × A is known as the universal relation.
2. ∅ is known as the empty relation.

Binary Relation on a Set

- Let A be the set {1, 2, 3, 4}, which ordered pairs are in the relation R = {(a, b) | a divides b}?

- **R** = { (1,1), (1,2), (1,3), (1,4), (2,2), (2,4), (3,3), (4,4)}



$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Relations and Their Properties

Binary Relation on a Set (*cont.*)

Question: How many different relations are there on a set A with n elements?

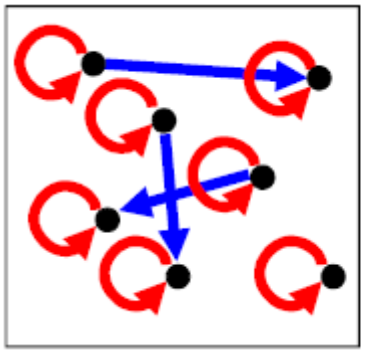
Solution:

- Suppose A has n elements
- Recall, a relation on a set A is a subset of A x A
- A x A has n2 n^2elements
- If a set has m element, its has 2^m subsets
- Therefore, the answer is 2^{n^2}.

Reflexive Relations

Definition: R is *reflexive* iff $(a,a) \in R$ for every element $a \in A$. Written symbolically, R is reflexive if and only if

$$\forall a[a \in U \longrightarrow (a,a) \in R]$$



Reflexive

$\forall a\ ((a, a) \in R)$

Every node has a self-loop

$$\begin{bmatrix} 1 & ? & \\ & 1 & \\ ? & & 1 \\ & & & 1 \end{bmatrix}$$

Reflexive

$\forall a\ ((a, a) \in R)$

All 1's on diagonal

If $A = \emptyset$ then the empty relation is reflexive vacuously. That is the empty relation on an empty set is reflexive!

Reflexive Relations

EXAMPLE: Let A = {1, 2, 3, 4} and determine whether relations R1, R2, R3, and R4 are Reflexive?

$R1 = \{(1, 1), (3, 3), (2, 2), (4, 4)\}$

$R2 = \{(1, 1), (1, 4), (2, 2), (3, 3), (4, 3)\}$

$R3 = \{(1, 1), (1, 2), (2, 1), (2, 2), (3, 3), (4, 4)\}$

$R4 = \{(1, 3), (2, 2), (2, 4), (3, 1), (4, 4)\}$

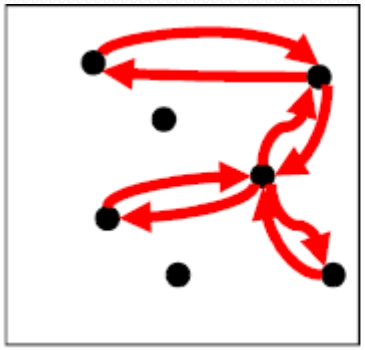
Solution:

- R1 is reflexive, since $(a, a) \in R1$ for all $a \in A$.
- R2 is not reflexive, because $(4, 4) \notin R2$.
- R3 is reflexive, since $(a, a) \in R3$ for all $a \in A$.
- R4 is not reflexive, because $(1, 1) \notin R4$, $(3, 3) \notin R4$

Symmetric Relations

Definition: R is *symmetric* iff $(b,a) \in R$ whenever $(a,b) \in R$ for all $a,b \in A$. Written symbolically, R is symmetric if and only if

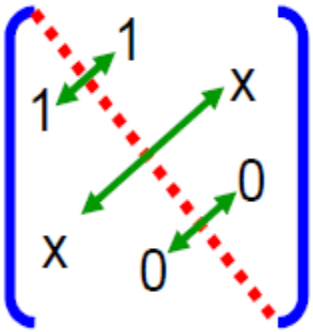
$$\forall a \forall b\, [(a,b) \in R \longrightarrow (b,a) \in R]$$



Symmetric

$\forall a\ \forall b\ (((a, b) \in R) \rightarrow ((b, a) \in R))$

Every link is bidirectional



Symmetric

$\forall a\ \forall b\ (((a, b) \in R) \rightarrow ((b, a) \in R))$

All identical across diagonal

Accordingly, R is symmetric if the elements in the ith row are the same as the elements in the ith column of the matrix M representing R. More precisely, M is a symmetric matrix
i.e. M = M^t

Symmetric Relations

EXAMPLE: Let A = {1, 2, 3, 4} and determine whether relations R1, R2, R3, and R4 are Symmetric?

$R1 = \{(1, 1), (1, 3), (2, 4), (3, 1), (4,2)\}$

$R2 = \{(1, 1), (2, 2), (3, 3), (4, 4)\}$

$R3 = \{(2, 2), (2, 3), (3, 4)\}$

$R4 = \{(1, 1), (2, 2), (3, 3), (4, 3), (4, 4)\}$

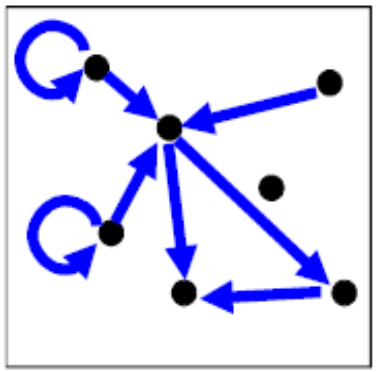
Solution:

- R1 is Symmetric, since (a, b) and (b, a) $\in$R1 for all (a, b) $\in$A.
- R2 is also symmetric. We say it is vacuously true.
- R3 is not symmetric, because (2,3) $\in$ R3 but (3,2) $\notin$ R3.
- R4 is not symmetric because (4,3) $\in$ R4 but (3,4) $\notin$ R4.

Antisymmetric Relations

Definition: A relation R on a set A such that for all $a,b \in A$ if $(a,b) \in R$ and $(b,a) \in R$, then $a = b$ is called *antisymmetric*. Written symbolically, R is antisymmetric if and only if $\forall a \forall b\,[(a,b) \in R \wedge (b,a) \in R \longrightarrow a = b]$

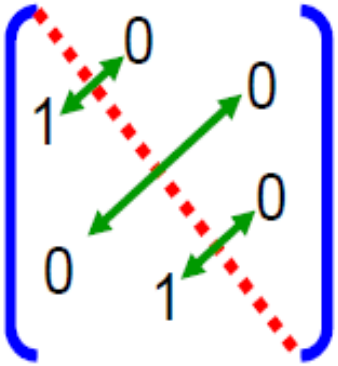
Note: (a,a) may be an element in R.



Antisymmetric

$\forall a\ \forall b\ (((a, b) \in R \wedge (b, a) \in R) \rightarrow (a = b))$

No link is bidirectional



Antisymmetric

$\forall a\ \forall b\ (((a, b) \in R \wedge (b, a) \in R) \rightarrow (a = b))$

All 1's are across from 0's

Let R be an anti-symmetric relation on a set A = {a1, a2, …, an}. Then if (ai, aj) ∈R for i ≠ j then (ai, aj) ∉R. Thus in the matrix representation of R there is a 1 in the ith row and jth column iff the jth row and ith column contains o vice versa.

Antisymmetric Relations

EXAMPLE: Let A = {1, 2, 3, 4} and determine whether relations R1, R2, R3, and R4 are Antisymmetric?

R1 = {(1,1),(2,2),(3,3)}

R2 = {(1,2),(2,2), (2,3), (3,4), (4,1)}

R3={(1,3),(2,2), (2,4), (3,1), (4,2)}

R4={(1,3),(2,4), (3,1), (4,3)}

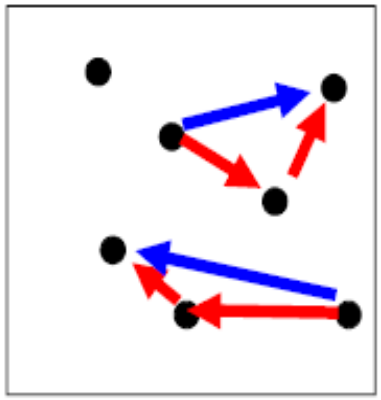
Solution:

- R1 is anti-symmetric and symmetric .
- R2 is anti-symmetric but not symmetric because (1,2) ∈ R2 but (2,1) ∉ R2.
- R3 is not anti-symmetric since (1,3) & (3,1) ∈ R3 but 1 ≠ 3. Note that R3 is symmetric.
- R4is neither anti-symmetric because (1,3) & (3,1) ∈ R4 but 1 ≠ 3 nor symmetric because (2,4) ∈ R4 but (4,2) ∉R4

Transitive Relations

Definition: A relation R on a set A is called transitive if whenever $(a,b) \in R$ and $(b,c) \in R$, then $(a,c) \in R$, for all $a,b,c \in A$. Written symbolically, R is transitive if and only if

$$\forall a \forall b \forall c[(a,b) \in R \wedge (b,c) \in \mathrm{R} \longrightarrow (a,c) \in R\ ]$$



Transitive

$\forall a \forall b \forall c\ (\ ((a,b) \in R \wedge (b,c) \in R) \rightarrow ((a,c) \in R))$

Every two adjacent forms a triangle
(Not easy to observe in Graph)

Transitive

$\forall a \forall b \forall c\ (\ ((a,b) \in R \wedge (b,c) \in R) \rightarrow ((a,c) \in R))$

Not easy to observe in Matrix

For a transitive directed graph, whenever there is an arrow going from one point to the second, and from the second to the third, there is an arrow going directly from the first to the third.

Transitive Relations

EXAMPLE: Let A = {1, 2, 3, 4} and determine whether relations R1, R2 and R3 are Transitive?

$R1 = \{(1, 1), (1, 2), (1, 3), (2, 3)\}$

$R2 = \{(1, 2), (1, 4), (2, 3), (3, 4)\}$

$R3 = \{(2, 1), (2, 4), (2, 3), (3,4)\}$

Solution:

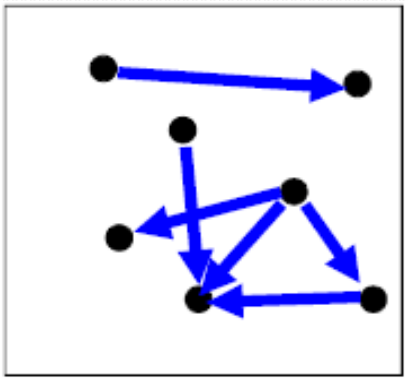
- R1 is transitive because (1, 1), (1, 2) are in R, then to be transitive relation(1,2) must be there and it belongs to R.
- R2 is not transitive since (1,2) and (2,3) $\in$ R2 but (1,3) $\notin$ R2.
- R3 is transitive.(check by definition)

Irreflexive Relations

Definition: R is irreflexive iff for all a∈A,(a,a) ∉R. That is, R is irreflexive if no element in A is related to itself by R.

Written symbolically, R is irreflexive if and only if

$$\forall a\,[(a \in A) \longrightarrow (a\,,\,a) \notin R]$$

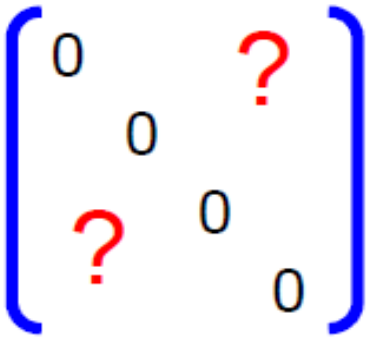


Irreflexive

$\forall a\,((a \in A) \rightarrow ((a, a) \notin R))$

No node links to itself

R is not irreflexive iff there is an element a∈A such that (a,a) ∈R.



Irreflexive

$\forall a\,((a \in A) \rightarrow ((a, a) \notin R))$

All 0's on diagonal

Irreflexive Relations

EXAMPLE: Let A = {1, 2, 3, 4} and determine whether relations R1, R2 and R3 are Irreflexive?

R1 = {(1,3), (1,4), (2,3), (2,4), (3,1), (3,4)}

R2 = {(1,1), (1,2), (2,1), (2,2), (3,3), (4,4)}

R3= {(1,2), (2,3), (3,3), (3,4)}

Solution:

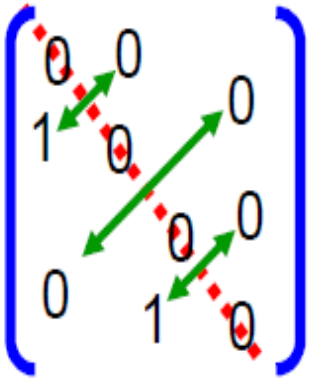
- R1 is irreflexive since no element of A is related to itself in R1. i.e. (1,1)$\notin$ R1, (2,2) $\notin$ R1, (3,3) $\notin$ R1,(4,4) $\notin$ R1.
- R2 is not irreflexive, since all elements of A are related to themselves in R2.
- R3 is not irreflexive since (3,3) $\in$R3. Note that R3 is not reflexive.

Asymmetric Relations

Definition: R is Asymmetric iff for all $(a,b) \in R$ *than* $(b,a) \notin R$.
Written symbolically, R is Asymmetric if and only if

$$\forall a \forall b[((a,b) \in R)) \longrightarrow ((b,a) \notin R)]$$

Note:(a,a) cannot be an element in R.

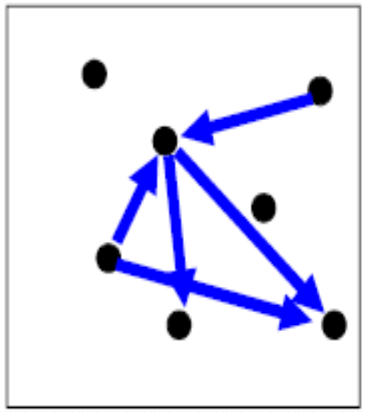


Asymmetric

$\forall a \forall b (((a, b) \in R) \rightarrow ((b, a) \notin R))$

All 1's are across from 0's (Antisymmetric)
All 0's on diagonal (Irreflexive)

Asymmetry = Antisymmetry + Irreflexivity



Asymmetric

$\forall a \forall b (((a, b) \in R) \rightarrow ((b, a) \notin R))$

No link is bidirectional (Antisymmetric)
No node links to itself (Irreflexive)

Asymmetric Relations

- **EXAMPLE**: Let A = {1, 2, 3, 4} and determine whether relations R1, R2 and R3 are Asymmetric?

  R1 = {(1,1), (1,2), (2,1), (2,2), (3,4), (4,1), (4,4)}

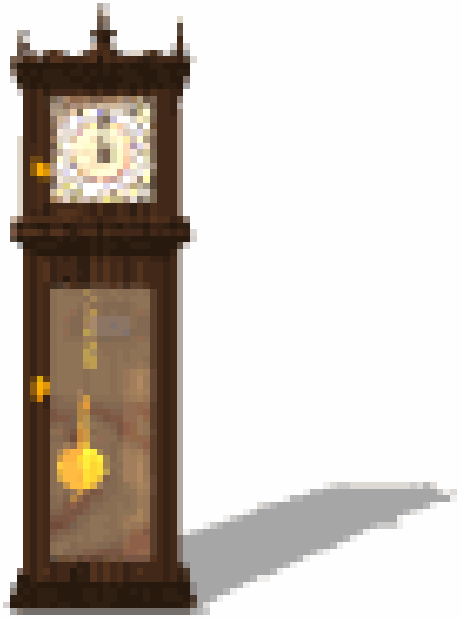
  R2 = {(1,2),(2,3),(3,4)}

  R3 = {(2,3),(3,3),(3,4)}

Solution:

- R1 is not Asymmetric since R1 in neither Antisymmetric nor Irreflexive.
- R2 is Asymmetric since R2 is both Antisymmetric and Irreflexive.
- R3 is not Asymmetric since it is Antisymmetric but not irreflexive.

Activity Time



Consider the following relations on {1, 2, 3, 4}:

$R1 = \{(1, 1), (1, 2), (2, 1), (2, 2), (3, 4), (4, 1), (4, 4)\}$,

$R2 = \{(1, 1), (1, 2), (2, 1)\}$,

$R3 = \{(1, 1), (1, 2), (1, 4), (2, 1), (2, 2), (3, 3), (4, 1), (4, 4)\}$,

$R4 = \{(2, 1), (3, 1), (3, 2), (4, 1), (4, 2), (4, 3)\}$,

$R5 = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 3), (3, 4), (4, 4)\}$,

$R6 = \{(3, 4)\}$.

Determine which of these relation are Reflexive, Symmetric, Transitive, Antisymmetric, Irreflexive and Asymmetric.

Combining Relations

As R is a subsets of A x B, the set operations can be applied

- Union (U)
- Intersection ($\cap$)
- Difference (-)
- Symmetric Complement ($\oplus$)

Given two relations R_1 and R_2, we can combine them using basic set operations to form new relations such as $R_1 \cup R_2$, $R_1 \cap R_2$, $R_1 - R_2$, $R_2 - R_1$ and $R_1 \oplus R_2$.

Combining Relations

Given, $A = \{1,2,3\}$, $B = \{1,2,3,4\}$

$R_1 = \{(1,1),(2,2),(3,3)\}$,

$R_2 = \{(1,1),(1,2),(1,3),(1,4)\}$

- $R_1 \cup R_2 = \{(1,1),(1,2),(1,3),(1,4),(2,2),(3,3)\}$
- $R_1 \cap R_2 = \{(1,1)\}$
- $R_1 - R_2 = \{(2,2),(3,3)\}$
- $R_2 - R_1 = \{(1,2),(1,3),(1,4)\}$
- $R_1 \oplus R_2 = \{(1,2),(1,3),(1,4),(2,2),(3,3)\}$

Composition of Relations

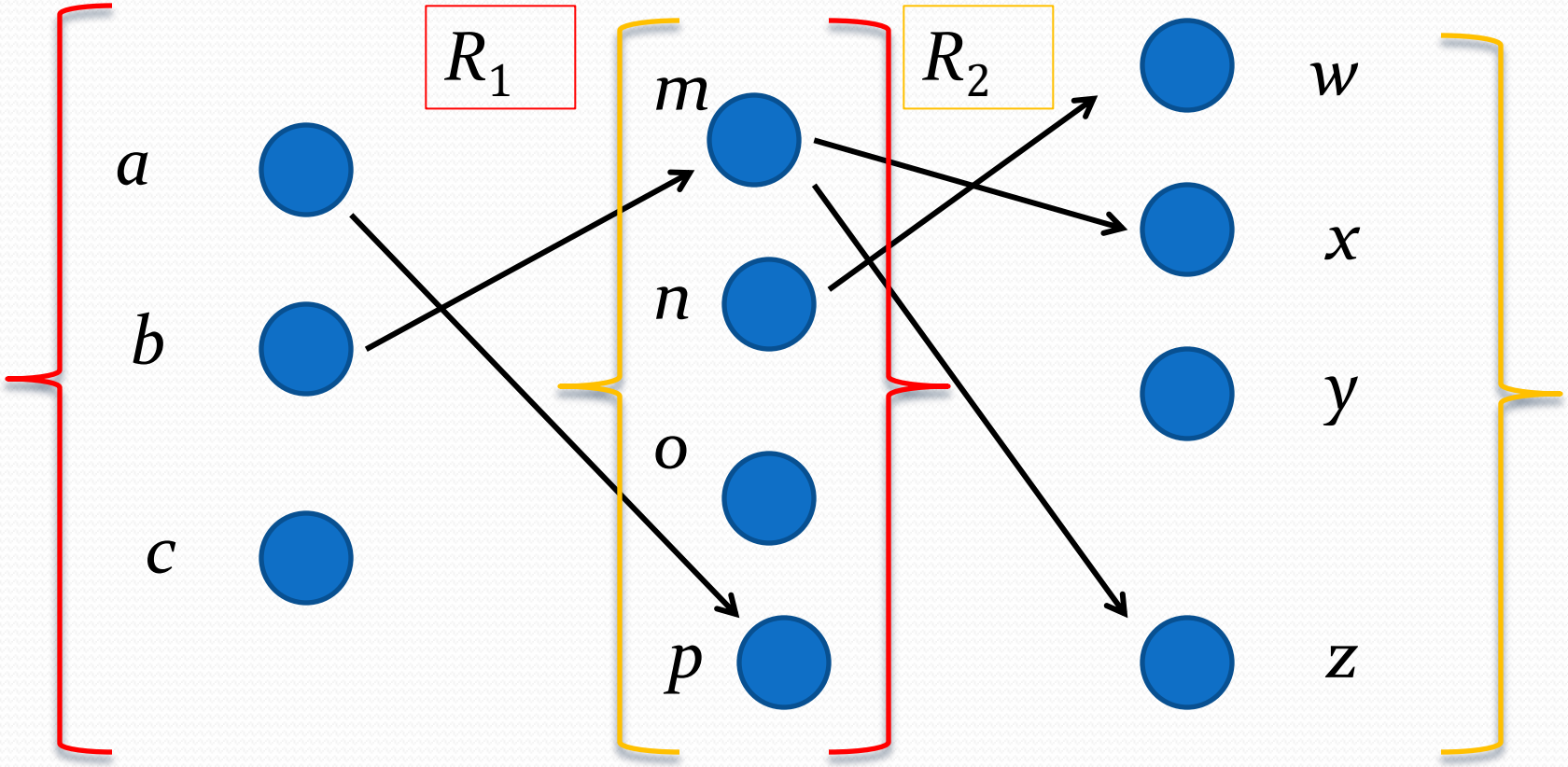
Definition: Suppose

- R_1 is a relation from a set A to a set B.
- R_2 is a relation from B to a set C.

Then the *composition* (or *composite*) of R_2 with R_1, is a relation from A to C where

- if (x,y) is a member of R_1 and (y,z) is a member of R_2, then (x,z) is a member of $R_2 \circ R_1$.

Representing the Composition of a Relation



$$R_1 \circ R_2 = \{(b,D),(b,B)\}$$

Composition of Relations

What is the composite of the relations R and S, where

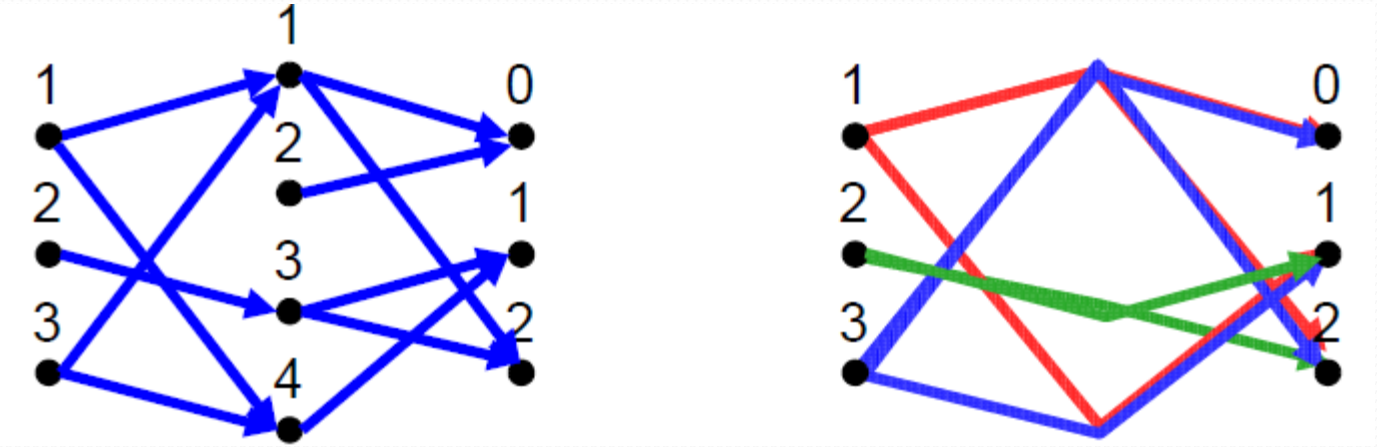
- R is the relation from **{1,2,3}** to **{1,2,3,4}** with

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

- S is the relation from **{1,2,3,4}** to **{0,1,2}** with

  S = {(1,0),(1,2),(2,0),(3,1),(3,2),(4,1)}?

- S o R = {(1,0),(1,2),(1,1), (2,2),((2,1), 3,0),(3,2),(3,1)}



INVERSE OF A RELATION

Let R be a relation from A to B. The inverse relation R^{-1} from B to A is defined as:

$$R^{-1} = \{(b,a) \in B \times A \mid (a,b) \in R\}$$

More simply, the inverse relation R^{-1} of R is obtained by interchanging the elements of all the ordered pairs in R.

- **Example**

X = {a, b, c} and Y={1, 2}

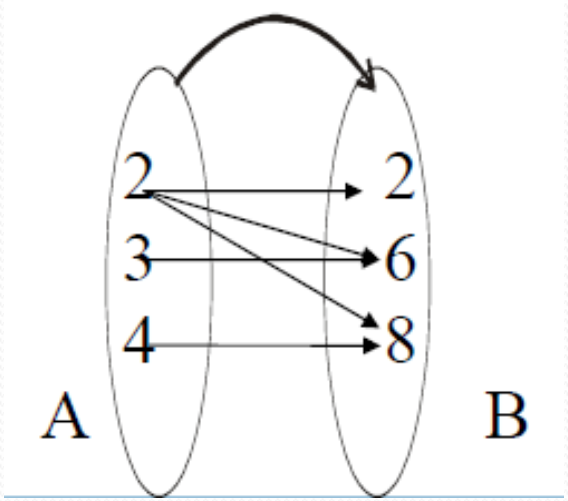
R = {(a, 1), (b, 2), (c, 1)}

- R^{-1} = {(1, a), (2, b), (1, c)}

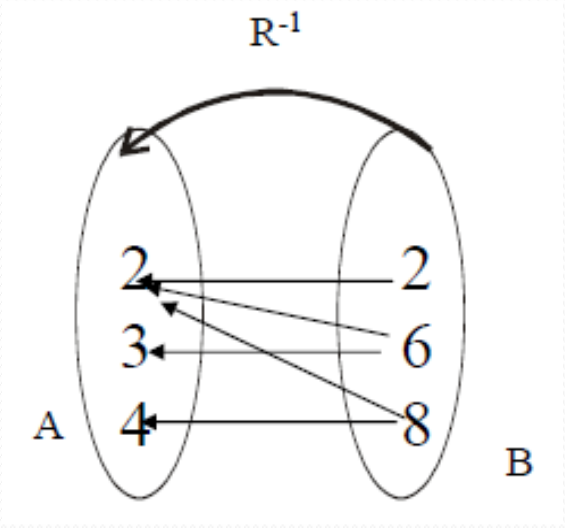
INVERSE OF A RELATION

The relation

R = {(2,2), (2,6), (2,8), (3,6), (4,8)} is represented by the arrow diagram.



Then inverse of the above relation can be obtained simply changing the directions of the arrows and hence the diagram is



Equivalence Relations

Equivalence Relations

Definition 1: A relation on a set A is called an *equivalence relation* if it is reflexive, symmetric, and transitive.

Definition 2: Two elements a, and b that are related by an equivalence relation are called *equivalent*. The notation $a \sim b$ is often used to denote that a and b are equivalent elements with respect to a particular equivalence relation.

Strings

Example:

Suppose that R is the relation on the set of strings 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: Show that all of the properties of an equivalence relation hold.

- *Reflexivity*: Because $l(a) = l(a)$, it follows that aRa for all strings a.
- *Symmetry*: Suppose that aRb. Since $l(a) = l(b)$, $l(b) = l(a)$ also holds and bRa.
- *Transitivity*: Suppose that aRb and bRc. Since $l(a) = l(b)$, and $l(b) = l(c)$, $l(a) = l(a)$ also holds and aRc.

Congruence Modulo m

Example: Let m be an integer with $m > 1$. Show that the relation

$R = \{(a,b) \mid a \equiv b \pmod{m}\}$

is an equivalence relation on the set of integers.

Solution: Recall that $a \equiv b \pmod{m}$ if and only if m divides $a - b$.

- *Reflexivity*: $a \equiv a \pmod{m}$ since $a - a = 0$ is divisible by m since $0 = 0 \cdot m$.
- *Symmetry*: Suppose that $a \equiv b \pmod{m}$. Then $a - b$ is divisible by m, and so $a - b = km$, where k is an integer. It follows that $b - a = (-k)\, m$, so $b \equiv a \pmod{m}$.
- *Transitivity*: Suppose that $a \equiv b \pmod{m}$ and $b \equiv c \pmod{m}$. Then m divides both $a - b$ and $b - c$. Hence, there are integers k and l with $a - b = km$ *and* $b - c = lm$. We obtain by adding the equations:

  $$a - c = (a - b) + (b - c) = km + lm = (k + l)\, m.$$

  Therefore, $a \equiv c \pmod{m}$.

Divides

Example: Show that the "divides" relation on the set of positive integers is not an equivalence relation.

Solution: The properties of reflexivity, and transitivity do hold, but there relation is not transitive. Hence, "divides" is not an equivalence relation.

- *Reflexivity*: $a \mid a$ for all a.
- *Not Symmetric*: For example, $2 \mid 4$, but $4 \nmid 2$. Hence, the relation is not symmetric.
- *Transitivity*: Suppose that a divides b and b divides c. Then there are positive integers k and l such that $b = ak$ and $c = bl$. Hence, $c = a(kl)$, so a divides c. Therefore, the relation is transitive.

Partial Orderings

Partial Orderings

Definition 1: A relation R on a set S is called a *partial ordering,* or *partial order,* if it is reflexive, antisymmetric, and transitive.

A set 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 (*continued*)

Example 1: Show that the "greater than or equal" relation ($\geq$) is a partial ordering on the set of integers.

- *Reflexivity*: $a \geq a$ for every integer a.
- *Antisymmetry*: If $a \geq b$ and $b \geq a$, then $a = b$.
- *Transitivity*: If $a \geq b$ and $b \geq c$, then $a \geq c$.

These properties all follow from the order axioms for the integers. (*See Appendix* 1).

Partial Orderings (*continued*)

Example 2: Show that the divisibility relation (|) is a partial ordering on the set of integers.

- *Reflexivity*: $a \mid a$ for all integers a. (*see Example* 9 *in Section* 9.1)
- *Antisymmetry*: If a and b are positive integers with $a \mid b$ and $b \mid a$, then $a = b$. (*see Example* 12 *in Section* 9.1)
- *Transitivity*: Suppose that a divides b and b divides c. Then there are positive integers k and l such that $b = ak$ and $c = bl$. Hence, $c = a(kl)$, so a divides c. Therefore, the relation is transitive.

- $(\boldsymbol{Z}^+, |)$ is a poset.

Partial Orderings (*continued*)

Example 3: Show that the inclusion relation ($\subseteq$) is a partial ordering on the power set of a set S.

- *Reflexivity*: $A \subseteq A$ whenever A is a subset of S.
- *Antisymmetry*: If A and B are positive integers with $A \subseteq B$ and $B \subseteq A$, then $A = B$.
- *Transitivity*: If $A \subseteq B$ and $B \subseteq C$, then $A \subseteq C$.

The properties all follow from the definition of set inclusion.

Number Theory and Cryptography

Chapter 4

Chapter Motivation

- *Number theory* is the part of mathematics devoted to the study of the integers and their properties.
- Key ideas in number theory include divisibility and the primality of integers.
- Number theory has long been studied because of the beauty of its ideas, its accessibility, and its wealth of open questions.
- We'll use many ideas developed in Chapter 1 about proof methods and proof strategy in our exploration of number theory.
- Mathematicians have long considered number theory to be pure mathematics, but it has important applications to computer science and cryptography studied in Sections 4.5 and 4.6.

Chapter Summary

- Divisibility and Modular Arithmetic
- Primes and Greatest Common Divisors
- Solving Congruencies
- Applications of Congruencies
- Cryptography

Divisibility and Modular Arithmetic

Section 4.1

Section Summary

- Division
- Division Algorithm
- Modular Arithmetic

Division

Definition: If a and b are integers with $a \neq 0$, then a *divides* b if there exists an integer c such that $b = ac$.

- When a divides b we say that a is a *factor* or *divisor* of b and that b is a multiple of a.
- The notation $a \mid b$ denotes that a divides b.
- If $a \mid b$, then b/a is an integer.
- If a does not divide b, we write $a \nmid b$.

Example: Determine whether $3 \mid 7$ and whether $3 \mid 12$.

Properties of Divisibility

Theorem 1: Let a, b, and c be integers, where $a \neq 0$.

i. If $a \mid b$ and $a \mid c$, then $a \mid (b + c)$;

ii. If $a \mid b$, then $a \mid bc$ for all integers c;

iii. If $a \mid b$ and $b \mid c$, then $a \mid c$.

Proof: (i) Suppose $a \mid b$ and $a \mid c$, then it follows that there are integers s and t with $b = as$ and $c = at$. Hence,

$b + c = as + at = a(s + t)$. Hence, $a \mid (b + c)$

(Exercises 3 and 4 ask for proofs of parts (ii) and (iii).) ◄

Corollary: If a, b, and c be integers, where $a \neq 0$, such that $a \mid b$ and $a \mid c$, then $a \mid mb + nc$ whenever m and n are integers.

Can you show how it follows easily from from (ii) and (i) of Theorem 1?

Division Algorithm

- When an integer is divided by a positive integer, there is a quotient and a remainder. This is traditionally called the "Division Algorithm," but is really a theorem.

Division Algorithm: If a is an integer and d a positive integer, then there are unique integers q and r, with $0 \leq r < d$, such that $a = dq + r$ (*proved in Section* 5.2).

- d is called the *divisor*.
- a is called the *dividend*.
- q is called the *quotient*.
- r is called the *remainder*.

Definitions of Functions **div** and **mod**

$q = a$ **div** d

$r = a$ **mod** d

Examples:

- What are the quotient and remainder when 101 is divided by 11?
  Solution: The quotient when 101 is divided by 11 is 9 = 101 **div** 11, and the remainder is 2 = 101 **mod** 11.
- What are the quotient and remainder when −11 is divided by 3?
  Solution: The quotient when −11 is divided by 3 is −4 = −11 **div** 3, and the remainder is 1 = −11 **mod** 3.

Congruence Relation

Definition: If a and b are integers and m is a positive integer, then a is *congruent* to b *modulo* m if m divides $a - b$.

- The notation $a \equiv b \pmod{m}$ says that a is congruent to b modulo m.
- We say that $a \equiv b \pmod{m}$ is a *congruence* and that m is its *modulus*.
- Two integers are congruent mod m if and only if they have the same remainder when divided by m.
- If a is not congruent to b modulo m, we write

$$a \not\equiv b \pmod{m}$$

Example: Determine whether 17 is congruent to 5 modulo 6 and whether 24 and 14 are congruent modulo 6.

Solution:

- $17 \equiv 5 \pmod{6}$ because 6 divides $17 - 5 = 12$.
- $24 \not\equiv 14 \pmod{6}$ since 6 divides $24 - 14 = 10$ is not divisible by 6.

More on Congruences

Theorem 4: Let m be a positive integer. The integers a and b are congruent modulo m if and only if there is an integer k such that $a = b + km$.

Proof:

- If $a \equiv b \pmod{m}$, then (by the definition of congruence) $m \mid a - b$. Hence, there is an integer k such that $a - b = km$ and equivalently $a = b + km$.
- Conversely, if there is an integer k such that $a = b + km$, then $km = a - b$. Hence, $m \mid a - b$ and $a \equiv b \pmod{m}$. ◀

The Relationship between (mod m) and **mod** m Notations

- The use of "mod" in $a \equiv b \pmod{m}$ and $a \bmod m = b$ are different.
  - $a \equiv b \pmod{m}$ is a relation on the set of integers.
  - In $a \bmod m = b$, the notation **mod** denotes a function.
- The relationship between these notations is made clear in this theorem.
- **Theorem 3**: Let a and b be integers, and let m be a positive integer. Then $a \equiv b \pmod{m}$ if and only if $a \bmod m = b \bmod m$. (*Proof in the exercises*)

Congruencies of Sums and Products

Let m be a positive integer. If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then

$$a + c \equiv b + d \pmod{m} \text{ and } ac \equiv bd \pmod{m}$$

Example: Because $7 \equiv 2 \pmod{5}$ and $11 \equiv 1 \pmod{5}$, it follows from Theorem 5 that

$$18 = 7 + 11 \equiv 2 + 1 = 3 \pmod{5}$$

$$77 = 7\ 11 \equiv 2 + 1 = 3 \pmod{5}$$

◀

Algebraic Manipulation of Congruencies

- Multiplying both sides of a valid congruence by an integer preserves validity.

  If $a \equiv b \pmod{m}$ holds then $c \cdot a \equiv c \cdot b \pmod{m}$, where c is any integer, holds by Theorem 5 with $d = c$.

- Adding an integer to both sides of a valid congruence preserves validity.

  If $a \equiv b \pmod{m}$ holds then $c + a \equiv c + b \pmod{m}$, where c is any integer, holds by Theorem 5 with $d = c$.

- Dividing a congruence by an integer does not always produce a valid congruence.

  Example: The congruence $14 \equiv 8 \pmod{6}$ holds. But dividing both sides by 2 does not produce a valid congruence since $14/2 = 7$ and $8/2 = 4$, but $7 \not\equiv 4 \pmod{6}$.

Computing the **mod** *m* Function of Products and Sums

- We use the following corollary to Theorem 5 to compute the remainder of the product or sum of two integers when divided by m from the remainders when each is divided by m.

Corollary: Let m be a positive integer and let a and b be integers. Then

$$(a + b) \ (\mathbf{mod}\ m) = ((a \ \mathbf{mod}\ m) + (b \ \mathbf{mod}\ m)) \ \mathbf{mod}\ m$$

and

$$ab \ \mathbf{mod}\ m = ((a \ \mathbf{mod}\ m) \ (b \ \mathbf{mod}\ m)) \ \mathbf{mod}\ m.$$

(*proof in text*)

Applications of Congruences

Section Summary

- Hashing Functions
- Pseudorandom Numbers
- Check Digits

Hashing Functions

Definition: A *hashing function h* assigns memory location $h(k)$ to the record that has k as its key.

- A common hashing function is $h(k) = k$ **mod** m, where m is the number of memory locations.
- Because this hashing function is onto, all memory locations are possible.

Example: Let $h(k) = k$ **mod** 111. This hashing function assigns the records of customers with social security numbers as keys to memory locations in the following manner:

h(064212848) = 064212848 **mod** 111 = 14

h(037149212) = 037149212 **mod** 111 = 65

h(107405723) = 107405723 **mod** 111 = 14, but since location 14 is already occupied, the record is assigned to the next available position, which is 15.

- The hashing function is not one-to-one as there are many more possible keys than memory locations. When more than one record is assigned to the same location, we say a *collision* occurs. Here a collision has been resolved by assigning the record to the first free location.
- For collision resolution, we can use a *linear probing function*:

  $h(k,i) = (h(k) + i)$ **mod** m, where i runs from 0 to $m - 1$.
- There are many other methods of handling with collisions. You may cover these in a later CS course.

Pseudorandom Numbers

- Randomly chosen numbers are needed for many purposes, including computer simulations.
- *Pseudorandom numbers* are not truly random since they are generated by systematic methods.
- The *linear congruential method* is one commonly used procedure for generating pseudorandom numbers.
- Four integers are needed: the *modulus* m, the *multiplier* a, the *increment* c, and *seed* x_0, with $2 \leq a < m$, $0 \leq c < m$, $0 \leq x_0 < m$.
- We generate a sequence of pseudorandom numbers $\{x_n\}$, with $0 \leq x_n < m$ for all n, by successively using the recursively defined function

$$x_{n+1} = (ax_n + c) \bmod m.$$

(*an example of a recursive definition, discussed in Section* 5.3*)*

- If psudorandom numbers between 0 and 1 are needed, then the generated numbers are divided by the modulus, x_n / m.

Pseudorandom Numbers

- **Example**: Find the sequence of pseudorandom numbers generated by the linear congruential method with modulus $m = 9$, multiplier $a = 7$, increment $c = 4$, and seed $x_0 = 3$.
- **Solution**: Compute the terms of the sequence by successively using the congruence $x_{n+1} = (7x_n + 4) \bmod 9$, with $x_0 = 3$.

$x_1 = 7x_0 + 4 \bmod 9 = 7 \cdot 3 + 4 \bmod 9 = 25 \bmod 9 = 7,$
$x_2 = 7x_1 + 4 \bmod 9 = 7 \cdot 7 + 4 \bmod 9 = 53 \bmod 9 = 8,$
$x_3 = 7x_2 + 4 \bmod 9 = 7 \cdot 8 + 4 \bmod 9 = 60 \bmod 9 = 6,$
$x_4 = 7x_3 + 4 \bmod 9 = 7 \cdot 6 + 4 \bmod 9 = 46 \bmod 9 = 1,$
$x_5 = 7x_4 + 4 \bmod 9 = 7 \cdot 1 + 4 \bmod 9 = 11 \bmod 9 = 2,$
$x_6 = 7x_5 + 4 \bmod 9 = 7 \cdot 2 + 4 \bmod 9 = 18 \bmod 9 = 0,$
$x_7 = 7x_6 + 4 \bmod 9 = 7 \cdot 0 + 4 \bmod 9 = 4 \bmod 9 = 4,$
$x_8 = 7x_7 + 4 \bmod 9 = 7 \cdot 4 + 4 \bmod 9 = 32 \bmod 9 = 5,$
$x_9 = 7x_8 + 4 \bmod 9 = 7 \cdot 5 + 4 \bmod 9 = 39 \bmod 9 = 3.$

The sequence generated is 3,7,8,6,1,2,0,4,5,3,7,8,6,1,2,0,4,5,3,...

It repeats after generating 9 terms.

- Commonly, computers use a linear congruential generator with increment $c = 0$. This is called a *pure multiplicative generator*. Such a generator with modulus $2^{31} - 1$ and multiplier $7^5 = 16{,}807$ generates $2^{31} - 2$ numbers before repeating.

Check Digits: UPCs

- A common method of detecting errors in strings of digits is to add an extra digit at the end, which is evaluated using a function. If the final digit is not correct, then the string is assumed not to be correct.

Example: Retail products are identified by their *Universal Product Codes* (*UPC*s). Usually these have 12 decimal digits, the last one being the check digit. The check digit is determined by the congruence:

$$3x_1 + x_2 + 3x_3 + x_4 + 3x_5 + x_6 + 3x_7 + x_8 + 3x_9 + x_{10} + 3x_{11} + x_{12} \equiv 0 \pmod{10}.$$

a. Suppose that the first 11 digits of the UPC are 79357343104. What is the check digit?

b. Is 041331021641 a valid UPC?

Solution:

a. $3\cdot 7 + 9 + 3\cdot 3 + 5 + 3\cdot 7 + 3 + 3\cdot 4 + 3 + 3\cdot 1 + 0 + 3\cdot 4 + x_{12} \equiv 0 \pmod{10}$

$21 + 9 + 9 + 5 + 21 + 3 + 12 + 3 + 3 + 0 + 12 + x_{12} \equiv 0 \pmod{10}$

$98 + x_{12} \equiv 0 \pmod{10}$

$x_{12} \equiv 0 \pmod{10}$ So, the check digit is 2.

b. $3\cdot 0 + 4 + 3\cdot 1 + 3 + 3\cdot 3 + 1 + 3\cdot 0 + 2 + 3\cdot 1 + 6 + 3\cdot 4 + 1 \equiv 0 \pmod{10}$

$0 + 4 + 3 + 3 + 9 + 1 + 0 + 2 + 3 + 6 + 12 + 1 = 44 \equiv 4 \not\equiv \pmod{10}$

Hence, 041331021641 is not a valid UPC.

Check Digits:ISBNs

Books are identified by an *International Standard Book Number* (ISBN-10), a 10 digit code. The first 9 digits identify the language, the publisher, and the book. The tenth digit is a check digit, which is determined by the following congruence

$$x_{10} \equiv \sum_{i=1}^{9} i x_i \pmod{11}.$$

The validity of an ISBN-10 number can be evaluated with the equivalent $\sum_{i=1}^{10} i x_i \equiv 0 \pmod{11}$.

a. Suppose that the first 9 digits of the ISBN-10 are 007288008. What is the check digit?

b. Is 084930149X a valid ISBN10?

Solution:

a. $X_{10} \equiv 1\cdot 0 + 2\cdot 0 + 3\cdot 7 + 4\cdot 2 + 5\cdot 8 + 6\cdot 8 + 7\cdot 0 + 8\cdot 0 + 9\cdot 8 \pmod{11}$.
$X_{10} \equiv 0 + 0 + 21 + 8 + 40 + 48 + 0 + 0 + 72 \pmod{11}$.
$X_{10} \equiv 189 \equiv 2 \pmod{11}$. Hence, $X_{10} = 2$.

X is used for the digit 10.

b. $1\cdot 0 + 2\cdot 8 + 3\cdot 4 + 4\cdot 9 + 5\cdot 3 + 6\cdot 0 + 7\cdot 1 + 8\cdot 4 + 9\cdot 9 + 10\cdot 10 =$
$0 + 16 + 12 + 36 + 15 + 0 + 7 + 32 + 81 + 100 = 299 \equiv 2 \not\equiv 0 \pmod{11}$
Hence, 084930149X is not a valid ISBN-10.

- A *single error* is an error in one digit of an identification number and a *transposition error* is the accidental interchanging of two digits. Both of these kinds of errors can be detected by the check digit for ISBN-10. (*see text for more details*)

Cryptography

Section Summary

- Classical Cryptography
- Cryptosystems
- Public Key Cryptography
- RSA Cryptosystem
- Cryptographic Protocols
- Primitive Roots and Discrete Logarithms

Caesar Cipher

Julius Caesar created secret messages by shifting each letter three letters forward in the alphabet (sending the last three letters to the first three letters.) For example, the letter B is replaced by E and the letter X is replaced by A. This process of making a message secret is an example of *encryption*.

Here is how the encryption process works:

- Replace each letter by an integer from $\mathbf{Z}_{26}$, that is an integer from 0 to 25 representing one less than its position in the alphabet.
- The encryption function is $f(p) = (p + 3)$ **mod** 26. It replaces each integer p in the set $\{0,1,2,\ldots,25\}$ by $f(p)$ in the set $\{0,1,2,\ldots,25\}$.
- Replace each integer p by the letter with the position $p + 1$ in the alphabet.

Example: Encrypt the message "MEET YOU IN THE PARK" using the Caesar cipher.

Solution: 12 4 4 19 24 14 20 8 13 19 7 4 15 0 17 10.

Now replace each of these numbers p by $f(p) = (p + 3)$ **mod** 26.

15 7 7 22 1 17 23 11 16 22 10 7 18 3 20 13.

Translating the numbers back to letters produces the encrypted message

"PHHW BRX LQ WKH SDUN."

Caesar Cipher

- To recover the original message, use $f^{-1}(p) = (p-3) \bmod 26$. So, each letter in the coded message is shifted back three letters in the alphabet, with the first three letters sent to the last three letters. This process of recovering the original message from the encrypted message is called *decryption*.
- The Caesar cipher is one of a family of ciphers called *shift ciphers*. Letters can be shifted by an integer k, with 3 being just one possibility. The encryption function is

  $$f(p) = (p + k) \bmod 26$$

  and the decryption function is

  $$f^{-1}(p) = (p-k) \bmod 26$$

  The integer k is called a *key*.

Shift Cipher

Example 1: Encrypt the message "STOP GLOBAL WARMING" using the shift cipher with $k = 11$.

Solution: Replace each letter with the corresponding element of $\mathbf{Z}_{26}$.

18 19 14 15 6 11 14 1 0 11 22 0 17 12 8 13 6.

Apply the shift $f(p) = (p + 11)$ **mod** 26, yielding

3 4 25 0 17 22 25 12 11 22 7 11 2 23 19 24 17.

Translating the numbers back to letters produces the ciphertext

"DEZA RWZMLW HLCXTYR."

Shift Cipher

Example 2: Decrypt the message "LEWLYPLUJL PZ H NYLHA ALHJOLY" that was encrypted using the shift cipher with $k = 7$.

Solution: Replace each letter with the corresponding element of $\mathbf{Z}_{26}$.

11 4 22 11 24 15 11 20 9 11 15 25 7 13 24 11 7 0 0 11 7 9 14 11 24.

Shift each of the numbers by $-k = -7$ modulo 26, yielding

4 23 15 4 17 8 4 13 2 4 8 18 0 6 17 4 0 19 19 4 0 2 7 4 17.

Translating the numbers back to letters produces the decrypted message

"EXPERIENCE IS A GREAT TEACHER."

Arithmetic Modulo *m*

Definitions: Let $\mathbf{Z}_m$ be the set of nonnegative integers less than m: $\{0,1, \ldots, m-1\}$

- The operation $+_m$ is defined as $a +_m b = (a + b)$ **mod** m. This is *addition modulo m*.
- The operation $\cdot_m$ is defined as $a \cdot_m b = (a \,.\, b)$ **mod** m. This is *multiplication modulo m*.
- Using these operations is said to be doing *arithmetic modulo m*.

Example: Find $7 +_{11} 9$ and $7 \cdot_{11} 9$.

Solution: Using the definitions above:

- $7 +_{11} 9 = (7 + 9)$ **mod** $11 = 16$ **mod** $11 = 5$
- $7 \cdot_{11} 9 = (7 \cdot 9)$ **mod** $11 = 63$ **mod** $11 = 8$

Arithmetic Modulo *m*

- The operations $+_m$ and $\cdot_m$ satisfy many of the same properties as ordinary addition and multiplication.
  - *Closure*: If a and b belong to $\mathbf{Z}_m$, then $a +_m b$ and $a \cdot_m b$ belong to $\mathbf{Z}_m$.
  - *Associativity*: If a, b, and c belong to $\mathbf{Z}_m$, then $(a +_m b) +_m c = a +_m (b +_m c)$ and $(a \cdot_m b) \cdot_m c = a \cdot_m (b \cdot_m c)$.
  - *Commutativity*: If a and b belong to $\mathbf{Z}_m$, then $a +_m b = b +_m a$ and $a \cdot_m b = b \cdot_m a$.
  - *Identity elements*: The elements 0 and 1 are identity elements for addition and multiplication modulo m, respectively.
    - If a belongs to $\mathbf{Z}_m$, then $a +_m 0 = a$ and $a \cdot_m 1 = a$.

continued →

Arithmetic Modulo *m*

- *Additive inverses*: If $a \neq 0$ belongs to $\mathbf{Z}_m$, then $m - a$ is the additive inverse of a modulo m and o is its own additive inverse.
  - $a +_m (m - a) = 0$ and $0 +_m 0 = 0$
- *Distributivity*: If a, b, and c belong to $\mathbf{Z}_m$, then
  - $a \cdot_m (b +_m c) = (a \cdot_m b) +_m (a \cdot_m c)$ and $(a +_m b) \cdot_m c = (a \cdot_m c) +_m (b \cdot_m c)$.
- Exercises 42-44 ask for proofs of these properties.
- Multiplicatative inverses have not been included since they do not always exist. For example, there is no multiplicative inverse of 2 modulo 6.
- (*optional*) Using the terminology of abstract algebra, $\mathbf{Z}_m$ with $+_m$ is a commutative group and $\mathbf{Z}_m$ with $+_m$ and $\cdot_m$ is a commutative ring.

Primes and Greatest Common Divisors

Section 4.3

Section Summary

- Prime Numbers and their Properties
- Greatest Common Divisors and Least Common Multiples
- The Euclidian Algorithm
- gcds as Linear Combinations

Primes

Definition: A positive integer *p* greater than 1 is called *prime* if the only positive factors of *p* are 1 and *p*. A positive integer that is greater than 1 and is not prime is called *composite*.

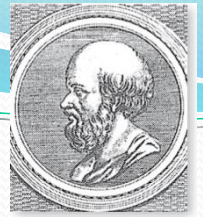
Example: The integer 7 is prime because its only positive factors are 1 and 7, but 9 is composite because it is divisible by 3.

The Fundamental Theorem of Arithmetic

Theorem: Every positive integer greater than 1 can be written uniquely as a prime or as the product of two or more primes where the prime factors are written in order of nondecreasing size.

Examples:

- $100 = 2 \cdot 2 \cdot 5 \cdot 5 = 2^2 \cdot 5^2$
- $641 = 641$
- $999 = 3 \cdot 3 \cdot 3 \cdot 37 = 3^3 \cdot 37$
- $1024 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^{10}$



Erastothenes
(276-194 B.C.)

The Sieve of Erastosthenes

- The *Sieve of Erastosthenes* can be used to find all primes not exceeding a specified positive integer. For example, begin with the list of integers between 1 and 100.
  a. Delete all the integers, other than 2, divisible by 2.
  b. Delete all the integers, other than 3, divisible by 3.
  c. Next, delete all the integers, other than 5, divisible by 5.
  d. Next, delete all the integers, other than 7, divisible by 7.
  e. Since all the remaining integers are not divisible by any of the previous integers, other than 1, the primes are:

{2,3,7,11,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89, 97}

continued →

The Sieve of Erastosthenes

TABLE 1 The Sieve of Eratosthenes.

Integers divisible by 2 other than 2 receive an underline.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Integers divisible by 3 other than 3 receive an underline.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Integers divisible by 5 other than 5 receive an underline.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

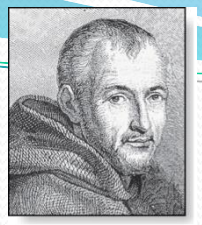
Integers divisible by 7 other than 7 receive an underline; integers in color are prime.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

If an integer n is a composite integer, then it has a prime divisor less than or equal to $\sqrt{n}$.

To see this, note that if $n = ab$, then $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$.

Trial division, a very inefficient method of determining if a number n is prime, is to try every integer $i \leq \sqrt{n}$ and see if n is divisible by i.



Marin Mersenne
(1588-1648)

Mersenne Primes

Definition: Prime numbers of the form $2^p - 1$, where p is prime, are called *Mersenne primes*.

- $2^2 - 1 = 3, 2^3 - 1 = 7, 2^5 - 1 = 37$, and $2^7 - 1 = 127$ are Mersenne primes.
- $2^{11} - 1 = 2047$ is not a Mersenne prime since $2047 = 23 \cdot 89$.
- There is an efficient test for determining if $2^p - 1$ is prime.
- The largest known prime numbers are Mersenne primes.
- As of mid 2011, 47 Mersenne primes were known, the largest is $2^{43,112,609} - 1$, which has nearly 13 million decimal digits.
- The *Great Internet Mersenne Prime Search* (*GIMPS*) is a distributed computing project to search for new Mersenne Primes.

http://www.mersenne.org/

Greatest Common Divisor

Definition: Let a and b be integers, not both zero. The largest integer d such that $d \mid a$ and also $d \mid b$ is called the greatest common divisor of a and b. The greatest common divisor of a and b is denoted by $\gcd(a,b)$.

One can find greatest common divisors of small numbers by inspection.

Example: What is the greatest common divisor of 24 and 36?

Solution: $\gcd(24,36) = 12$

Example: What is the greatest common divisor of 17 and 22?

Solution: $\gcd(17,22) = 1$

Greatest Common Divisor

Definition: The integers a and b are *relatively prime* if their greatest common divisor is 1.

Example: 17 and 22

Definition: The integers $a_1, a_2, \ldots, a_n$ are *pairwise relatively prime* if $\gcd(a_i, a_j) = 1$ whenever $1 \leq i < j \leq n$.

Example: Determine whether the integers 10, 17 and 21 are pairwise relatively prime.

Solution: Because $\gcd(10,17) = 1$, $\gcd(10,21) = 1$, and $\gcd(17,21) = 1$, 10, 17, and 21 are pairwise relatively prime.

Example: Determine whether the integers 10, 19, and 24 are pairwise relatively prime.

Solution: Because $\gcd(10,24) = 2$, 10, 19, and 24 are not pairwise relatively prime.

Finding the Greatest Common Divisor Using Prime Factorizations

- Suppose the prime factorizations of a and b are:

$$a = p_1^{a_1} p_2^{a_2} \cdots p_n^{a_n} , \qquad b = p_1^{b_1} p_2^{b_2} \cdots p_n^{b_n} ,$$

where each exponent is a nonnegative integer, and where all primes occurring in either prime factorization are included in both. Then:

$$\gcd(a, b) = p_1^{\min(a_1, b_1)} p_2^{\min(a_2, b_2)} \cdots p_n^{\min(a_n, b_n)} .$$

- This formula is valid since the integer on the right (of the equals sign) divides both a and b. No larger integer can divide both a and b.

Example: $120 = 2^3 \cdot 3 \cdot 5 \qquad 500 = 2^2 \cdot 5^3$

$\gcd(120,500) = 2^{\min(3,2)} \cdot 3^{\min(1,0)} \cdot 5^{\min(1,3)} = 2^2 \cdot 3^0 \cdot 5^1 = 20$

- Finding the gcd of two positive integers using their prime factorizations is not efficient because there is no efficient algorithm for finding the prime factorization of a positive integer.

Least Common Multiple

Definition: The least common multiple of the positive integers a and b is the smallest positive integer that is divisible by both a and b. It is denoted by $\text{lcm}(a,b)$.

- The least common multiple can also be computed from the prime factorizations.

$$\text{lcm}(a,b) = p_1^{\max(a_1,b_1)} p_2^{\max(a_2,b_2)} \cdots p_n^{\max(a_n,b_n)}$$

This number is divided by both a and b and no smaller number is divided by a and b.

Example: $\text{lcm}(2^3 3^5 7^2, 2^4 3^3) = 2^{\max(3,4)} 3^{\max(5,3)} 7^{\max(2,0)} = 2^4 3^5 7^2$

- The greatest common divisor and the least common multiple of two integers are related by:

Theorem 5: Let a and b be positive integers. Then

$$ab = \gcd(a,b) \cdot \text{lcm}(a,b)$$

(*proof is Exercise* 31)

Euclidean Algorithm

Euclid
(325 B.C.E. – 265 B.C.E.)

- The Euclidian algorithm is an efficient method for computing the greatest common divisor of two integers. It is based on the idea that gcd(a,b) is equal to gcd(a,c) when $a > b$ and c is the remainder when a is divided by b.

Example: Find gcd(91, 287):

- $287 = 91 \cdot 3 + 14$
- $91 = 14 \cdot 6 + 7$
- $14 = 7 \cdot 2 + 0$

Divide 287 by 91

Divide 91 by 14

Divide 14 by 7

Stopping condition

$$\gcd(287, 91) = \gcd(91, 14) = \gcd(14, 7) = 7$$

continued →

Euclidean Algorithm

- The Euclidean algorithm expressed in pseudocode is:

```
procedure gcd(a, b: positive integers)
x := a
x := b
while   y ≠ 0
      r := x mod y
      x := y
      y := r
return x {gcd(a,b) is x}
```

- In Section 5.3, we'll see that the time complexity of the algorithm is $O(\log b)$, where $a > b$.

Étienne Bézout
(1730-1783)

gcds as Linear Combinations

Bézout's Theorem: If a and b are positive integers, then there exist integers s and t such that $\gcd(a,b) = sa + tb$.

Definition: If a and b are positive integers, then integers s and t such that $\gcd(a,b) = sa + tb$ are called *Bézout coefficients* of a and b. The equation $\gcd(a,b) = sa + tb$ is called *Bézout's identity*.

- By Bézout's Theorem, the gcd of integers a and b can be expressed in the form $sa + tb$ where s and t are integers. This is a *linear combination* with integer coefficients of a and b.
  - $\gcd(6,14) = (-2)\cdot 6 + 1\cdot 14$

Finding gcds as Linear Combinations

Example: Express gcd(252,198) = 18 as a linear combination of 252 and 198.

Solution: First use the Euclidean algorithm to show gcd(252,198) = 18

i. $252 = 1 \cdot 198 + 54$
ii. $198 = 3 \cdot 54 + 36$
iii. $54 = 1 \cdot 36 + 18$
iv. $36 = 2 \cdot 18$

- Now working backwards, from iii and i above
  - $18 = 54 - 1 \cdot 36$
  - $36 = 198 - 3 \cdot 54$
- Substituting the 2nd equation into the 1st yields:
  - $18 = 54 - 1 \cdot (198 - 3 \cdot 54) = 4 \cdot 54 - 1 \cdot 198$
- Substituting $54 = 252 - 1 \cdot 198$ (from i)) yields:
  - $18 = 4 \cdot (252 - 1 \cdot 198) - 1 \cdot 198 = 4 \cdot 252 - 5 \cdot 198$

- This method illustrated above is a two pass method. It first uses the Euclidian algorithm to find the gcd and then works backwards to express the gcd as a linear combination of the original two integers. A one pass method, called the *extended Euclidean algorithm*, is developed in the exercises.

Dividing Congruencies by an Integer

- Dividing both sides of a valid congruence by an integer does not always produce a valid congruence (see Section 4.1).
- But dividing by an integer relatively prime to the modulus does produce a valid congruence:

  Theorem 7: Let m be a positive integer and let a, b, and c be integers. If $ac \equiv bc \pmod{m}$ and $\gcd(c,m) = 1$, then $a \equiv b \pmod{m}$.

  Proof: Since $ac \equiv bc \pmod{m}$, $m \mid ac - bc = c(a - b)$ by Lemma 2 and the fact that $\gcd(c,m) = 1$, it follows that $m \mid a - b$. Hence, $a \equiv b \pmod{m}$. ◀

Solving Congruencies

Section 4.4

Section Summary

- Linear Congruencies
- The Chinese Remainder Theorem
- Fermat's Little Theorem
- Pseudo primes

Linear Congruencies

Definition: A congruence of the form

$$ax \equiv b \pmod{m},$$

where m is a positive integer, a and b are integers, and x is a variable, is called a *linear congruence*.

- The solutions to a linear congruence $ax \equiv b \pmod{m}$ are all integers x that satisfy the congruence.

Definition: An integer $\bar{a}$ such that $\bar{a}a \equiv 1 \pmod{m}$ is said to be an *inverse* of a modulo m.

Example: 5 is an inverse of 3 modulo 7 since $5 \cdot 3 = 15 \equiv 1 \pmod{7}$

- One method of solving linear congruencies makes use of an inverse $\bar{a}$, if it exists. Although we can not divide both sides of the congruence by a, we can multiply by $\bar{a}$ to solve for x.

Inverse of a modulo m

- The following theorem guarantees that an inverse of a modulo m exists whenever a and m are relatively prime. Two integers a and b are relatively prime when $\gcd(a,b) = 1$.

Theorem 1: If a and m are relatively prime integers and $m > 1$, then an inverse of a modulo m exists. Furthermore, this inverse is unique modulo m. (This means that there is a unique positive integer $\bar{a}$ less than m that is an inverse of a modulo m and every other inverse of a modulo m is congruent to $\bar{a}$ modulo m.)

Proof: Since $\gcd(a,m) = 1$, by Theorem 6 of Section 4.3, there are integers s and t such that $sa + tm = 1$.

- Hence, $sa + tm \equiv 1 \pmod{m}$.
- Since $tm \equiv 0 \pmod{m}$, it follows that $sa \equiv 1 \pmod{m}$
- Consequently, s is an inverse of a modulo m.
- The uniqueness of the inverse is Exercise 7.

◀

Finding Inverses

- The Euclidean algorithm and Bézout coefficients gives us a systematic approaches to finding inverses.

  Example: Find an inverse of 3 modulo 7.

  Solution: Because gcd(3,7) = 1, by Theorem 1, an inverse of 3 modulo 7 exists.

  - Using the Euclidian algorithm: $7 = 2 \cdot 3 + 1$.
  - From this equation, we get $-2 \cdot 3 + 1 \cdot 7 = 1$, and see that -2 and 1 are Bézout coefficients of 3 and 7.
  - Hence, -2 is an inverse of 3 modulo 7.
  - Also every integer congruent to -2 modulo 7 is an inverse of 3 modulo 7, i.e., 5, -9, 12, etc.

Finding Inverses

Example: Find an inverse of 101 modulo 4620.

Solution: First use the Euclidian algorithm to show that gcd(101,4620) = 1.

$$42620 = 45 \cdot 101 + 75$$
$$101 = 1 \cdot 75 + 26$$
$$75 = 2 \cdot 26 + 23$$
$$26 = 1 \cdot 23 + 3$$
$$23 = 7 \cdot 3 + 2$$
$$3 = 1 \cdot 2 + 1$$
$$2 = 2 \cdot 1$$

Since the last nonzero remainder is 1, gcd(101,4260) = 1

Working Backwards:

$$1 = 3 - 1 \cdot 2$$
$$1 = 3 - 1 \cdot (23 - 7 \cdot 3) = -1 \cdot 23 + 8 \cdot 3$$
$$1 = -1 \cdot 23 + 8 \cdot (26 - 1 \cdot 23) = 8 \cdot 26 - 9 \cdot 23$$
$$1 = 8 \cdot 26 - 9 \cdot (75 - 2 \cdot 26) = 26 \cdot 26 - 9 \cdot 75$$
$$1 = 26 \cdot (101 - 1 \cdot 75) - 9 \cdot 75$$
$$= 26 \cdot 101 - 35 \cdot 75$$
$$1 = 26 \cdot 101 - 35 \cdot (42620 - 45 \cdot 101)$$
$$= -35 \cdot 42620 + 1601 \cdot 101$$

Bézout coefficients : – 35 and 1601

1601 is an inverse of 101 modulo 42620

Using Inverses to Solve Congruences

- We can solve the congruence $ax \equiv b \pmod{m}$ by multiplying both sides by $\bar{a}$.

Example: What are the solutions of the congruence $3x \equiv 4 \pmod{7}$.

Solution: We found that -2 is an inverse of 3 modulo 7 (two slides back). We multiply both sides of the congruence by -2 giving

$$-2 \cdot 3x \equiv -2 \cdot 4 \pmod{7}.$$

Because $-6 \equiv 1 \pmod{7}$ and $-8 \equiv 6 \pmod{7}$, it follows that if x is a solution, then $x \equiv -8 \equiv 6 \pmod{7}$

We need to determine if every x with $x \equiv 6 \pmod{7}$ is a solution. Assume that $x \equiv 6 \pmod{7}$. By Theorem 5 of Section 4.1, it follows that $3x \equiv 3 \cdot 6 = 18 \equiv 4 \pmod{7}$ which shows that all such x satisfy the congruence.

The solutions are the integers x such that $x \equiv 6 \pmod{7}$, namely, 6,13,20 … and $-1, -8, -15, \ldots$

The Chinese Remainder Theorem

Theorem 2: (*The Chinese Remainder Theorem*) Let $m_1, m_2, \ldots, m_n$ be pairwise relatively prime positive integers greater than one and $a_1, a_2, \ldots, a_n$ arbitrary integers. Then the system

$x \equiv a_1 \pmod{m_1}$

$x \equiv a_2 \pmod{m_2}$

.

.

.

$x \equiv a_n \pmod{m_n}$

has a unique solution modulo $m = m_1 m_2 \cdots m_n$.

(That is, there is a solution x with $0 \leq x < m$ and all other solutions are congruent modulo m to this solution.)

- **Proof**: We'll show that a solution exists by describing a way to construct the solution. Showing that the solution is unique modulo m is Exercise 30.

continued →

The Chinese Remainder Theorem

To construct a solution first let $M_k = m/m_k$ for $k = 1,2,\ldots,n$ and $m = m_1 m_2 \cdots m_n$.

Since $\gcd(m_k, M_k) = 1$, by Theorem 1, there is an integer y_k, an inverse of M_k modulo m_k, such that

$$M_k y_k \equiv 1 \pmod{m_k}.$$

Form the sum

$$x = a_1 M_1 y_1 + a_2 M_2 y_2 + \cdots + a_n M_n y_n \pmod{m}$$

Note that because $M_j \equiv 0 \pmod{m_k}$ whenever $j \neq k$, all terms except the kth term in this sum are congruent to 0 modulo m_k.

Because $M_k y_k \equiv 1 \pmod{m_k}$, we see that $x \equiv a_k M_k y_k \equiv a_k \pmod{m_k}$, for $k = 1,2,\ldots,n$.

Hence, x is a simultaneous solution to the n congruences.

$$x \equiv a_1 \pmod{m_1}$$
$$x \equiv a_2 \pmod{m_2}$$
$$\vdots$$
$$x \equiv a_n \pmod{m_n}$$

The Chinese Remainder Theorem

Example: Consider the 3 congruences from Sun-Tsu's problem:

$x \equiv 2 \pmod 3$, $x \equiv 3 \pmod 5$, $x \equiv 2 \pmod 7$.

- Let $m = 3 \cdot 5 \cdot 7 = 105$, $M_1 = m/3 = 35$, $M_3 = m/5 = 21$, $M_3 = m/7 = 15$.
- We see that
  - 2 is an inverse of $M_1 = 35$ modulo 3 since $35 \cdot 2 \equiv 2 \cdot 2 \equiv 1 \pmod 3$
  - 1 is an inverse of $M_2 = 21$ modulo 5 since $21 \equiv 1 \pmod 5$
  - 1 is an inverse of $M_3 = 15$ modulo 7 since $15 \equiv 1 \pmod 7$
- Hence,

$$x = a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3 \pmod m$$
$$= 2 \cdot 35 \cdot 2 + 3 \cdot 21 \cdot 1 + 2 \cdot 15 \cdot 1 = 233 \equiv 23 \pmod{105}$$

- We have shown that 23 is the smallest positive integer that is a simultaneous solution. Check it!

The Chinese Remainder Theorem

Word Problem:

Jessica breeds rabbits. She's not sure exactly how many she has today, but as she was moving them about this morning, she noticed some things. When she fed them, in groups of 5, she had 4 left over. When she bathed them, in groups of 8, she had a group of 6 left over. She took them outside to romp in groups of 9, but then the last group consisted of only 8. She's positive that there are fewer than 250 rabbits - but how many does she have?

Solution:

We have the following congurences

$x \equiv 4 \pmod 5$,

$x \equiv 6 \pmod 8$,

$x \equiv 8 \pmod 9$.

Fermat's Little Theorem

Pierre de Fermat
(1601-1665)

Theorem 3: (*Fermat's Little Theorem*) If p is prime and a is an integer not divisible by p, then $a^{p-1} \equiv 1 \pmod{p}$

Furthermore, for every integer a we have $a^p \equiv a \pmod{p}$

(*proof outlined in Exercise 19*)

Fermat's little theorem is useful in computing the remainders modulo p of large powers of integers.

Example: Find 7^{222} **mod** 11.

By Fermat's little theorem, we know that $7^{10} \equiv 1 \pmod{11}$, and so $(7^{10})^k \equiv 1 \pmod{11}$, for every positive integer k. Therefore,

$$7^{222} = 7^{22 \cdot 10 + 2} = (7^{10})^{22} 7^2 \equiv (1)^{22} \cdot 49 \equiv 5 \pmod{11}.$$

Hence, 7^{222} **mod** $11 = 5$.

Number Theory in Cryptography

Terminology: Two parties **Alice** and **Bob** want to communicate securely s.t. a third party **Eve** who intercepts messages cannot learn the content of the messages.

Symmetric Cryptosystems: Alice and Bob share a secret. Only they know a secret key K that is used to encrypt and decrypt messages. Given a message M, Alice encodes it (possibly with padding) into m, and then sends the ciphertext $encrypt(m, K)$ to Bob. Then Bob uses K to decrypt it and obtains $decrypt(encrypt(m, K), K) = m$.
Example: AES.

Public Key Cryptosystems: Alice and Bob do a-priori **not** share a secret. How can they establish a shared secret when others are listening to their messages?

Idea: Have a two-part key, i.e., a key pair. A public key that is used to encrypt messages, and a secret key to decrypt them. Alice uses Bob's public key to encrypt a message (everyone can do that). Only Bob can decrypt the message with his secret key.

Description of RSA: Key generation

- Choose two distinct prime numbers p and q. Numbers p and q should be chosen at random, and be of similar bit-length. Prime integers can be efficiently found using a primality test.
- Let $n = pq$ and $k = (p-1)(q-1)$. (In particular, $k = |Z_n^*|$).
- Choose an integer e such that $1 < e < k$ and $gcd(e, k) = 1$; i.e., e and k are coprime.
  e (for encryption) is released as the public key exponent.
  (e must not be very small.)
- Let d be the multiplicative inverse of e modulo k, i.e., $de \equiv 1 (\mod k)$. (Computed using the extended Euclidean algorithm.) d (for decryption) is the private key and kept secret.

The public key is (n, e) and the private key is (n, d).

RSA: Encryption and Decryption

Alice transmits her public key (n, e) to Bob and keeps the private key secret.

Encryption: Bob then wishes to send message M to Alice. He first turns M into an integer m, such that $0 \leq m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext c corresponding to

$$c \equiv m^e \mod n$$

This can be done quickly using the method of exponentiation by squaring. Bob then transmits c to Alice.

Decryption: Alice can recover m from c by using her private key exponent d via computing

$$m \equiv c^d \mod n$$

Given m, she can recover the original message M by reversing the padding scheme.

The RSA Algorithm

To generate a key pair:

- Pick large primes p and q (do not disclose them)
- Let n = p*q
- For the public key, choose e that is relatively prime to ø(n)=(p-1)(q-1).

public key = <e,n>

- For private key, find d that is the multiplicative inverse of e mod ø(n), i.e., e*d

Using RSA

Given pubKey = <e, n> and privKey = <d, n>

If Message = m

Then:

encryption: $c = m^e \bmod n$, $m < n$

decryption: $m = c^d \bmod n$

signature: $s = md \bmod n$, $m < n$

verification: $m = se \bmod n$

Example of RSA (1)

Choose p = 7 and q = 17.

Compute n = p*q= 119.

Compute f(n)=(p-1)(q-1)=96.

Select e = 5, (a relatively prime to f(n).)

Compute d = _77_such that e*d=1 mod f(n).

- Public key: <5,119>
- Private key: <77,119>
- Message = 19
- Encryption: 19^5 mod 119 = 66
- Decryption: 66^{77} mod 119 = 19

Example of RSA (2)

p = 7, q = 11, n = 77

Alice chooses e = 17, making d = 53

Bob wants to send Alice secret message

HELLO (07 04 11 11 14)

– 07^{17} mod 77 = 28; 04^{17} mod 77 = 16

– 11^{17} mod 77 = 44; – 11^{17} mod 77 = 44

– 14^{17} mod 77 = 42

- Bob sends 28 16 44 44 42

Example of RSA (3)

Alice receives **28 16 44 44 42**

Alice uses private key, d = 53, to decrypt message:

- 28^{53} mod 77 = 07; 16^{53} mod 77 = 04
- 44^{53} mod 77 = 11; 44^{53} mod 77 = 11
- 42^{53} mod 77 = 14

• Alice translates **07 04 11 11 14** to ***HELLO***

No one else could read it, as only Alice knows her private key (needed for decryption)

Graphs

Chapter 10

Chapter Summary

- Graphs and Graph Models
- Graph Terminology and Special Types of Graphs
- Representing Graphs and Graph Isomorphism
- Connectivity
- Euler and Hamiltonian Graphs
- Shortest-Path Problems
- Planar Graphs
- Graph Coloring

Graphs and Graph Models

Section 10.1

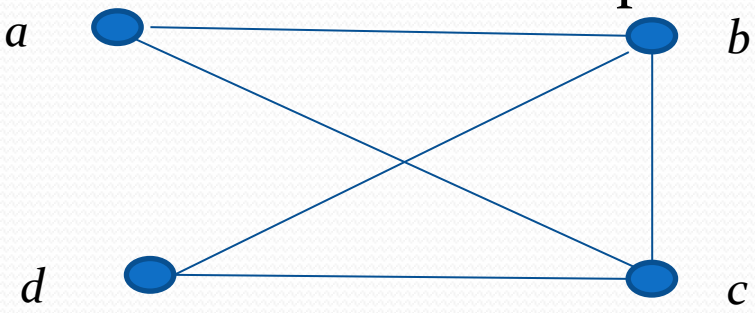
Section Summary

- Introduction to Graphs
- Graph Taxonomy
- Graph Models

Graphs

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

Example:



This is a graph with four vertices and five edges.

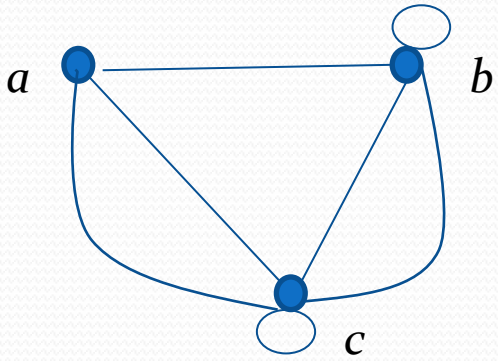
Remarks:

- The graphs we study here are unrelated to graphs of functions studied in Chapter 2.
- We have a lot of freedom when we draw a picture of a graph. All that matters is the connections made by the edges, not the particular geometry depicted. For example, the lengths of edges, whether edges cross, how vertices are depicted, and so on, do not matter
- A graph with an infinite vertex set is called an *infinite graph*. A graph with a finite vertex set is called a *finite graph*. We (following the text) restrict our attention to finite graphs.

Some Terminology

- In a *simple graph* each edge connects two different vertices and no two edges connect the same pair of vertices.
- *Multigraphs* may have multiple edges connecting the same two vertices. When m different edges connect the vertices u and v, we say that $\{u,v\}$ is an edge of *multiplicity* m.
- An edge that connects a vertex to itself is called a *loop*.
- A *pseudograph* may include loops, as well as multiple edges connecting the same pair of vertices.

Example:
This pseudograph has both multiple edges and a loop.



Remark: There is no standard terminology for graph theory. So, it is crucial that you understand the terminology being used whenever you read material about graphs.

Directed Graphs

Definition: An *directed graph* (or *digraph*) $G = (V, E)$ consists of a nonempty set V of *vertices* (or *nodes*) and a set E of *directed edges* (or *arcs*). Each edge is associated with an ordered pair of vertices. The directed edge associated with the ordered pair (u,v) is said to *start at* u and *end at* v.

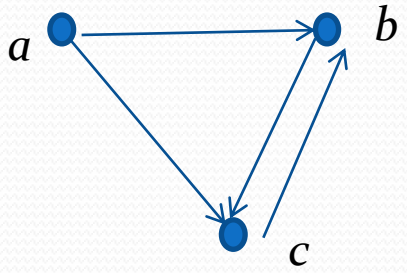
Remark:

- Graphs where the end points of an edge are not ordered are said to be *undirected graphs*.

Some Terminology (*continued*)

- A *simple directed graph* has no loops and no multiple edges.

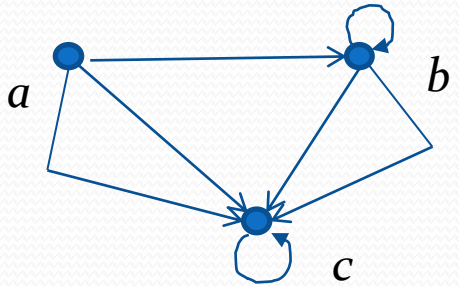
Example:
This is a directed graph with three vertices and four edges.



- A *directed multigraph* may have multiple directed edges. When there are m directed edges from the vertex u to the vertex v, we say that (u,v) is an edge of *multiplicity* m.

Example:

In this directed multigraph the multiplicity of (a,b) is 1 and the multiplicity of (b,c) is 2.



Graph Terminology: Summary

- To understand the structure of a graph and to build a graph model, we ask these questions:
  - Are the edges of the graph undirected or directed (or both)?
  - If the edges are undirected, are multiple edges present that connect the same pair of vertices? If the edges are directed, are multiple directed edges present?
  - Are loops present?

TABLE 1 Graph Terminology.

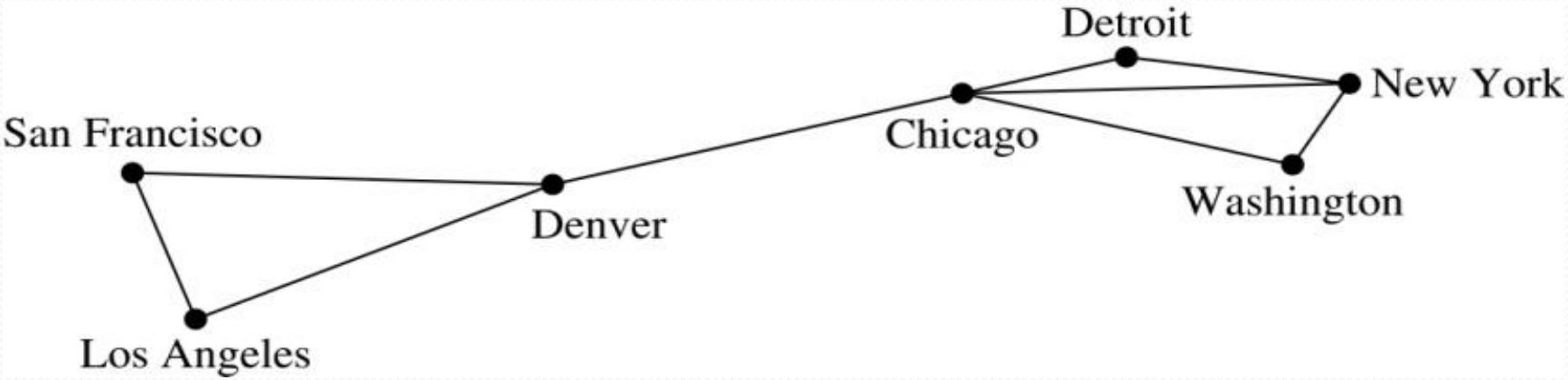
Type	*Edges*	*Multiple Edges Allowed?*	*Loops Allowed?*
Simple graph	Undirected	No	No
Multigraph	Undirected	Yes	No
Pseudograph	Undirected	Yes	Yes
Simple directed graph	Directed	No	No
Directed multigraph	Directed	Yes	Yes
Mixed graph	Directed and undirected	Yes	Yes

Other Applications of Graphs

- We will illustrate how graph theory can be used in models of:
  - Social networks
  - Communications networks
  - Information networks
  - Software design
  - Transportation networks
  - Biological networks
- It's a challenge to find a subject to which graph theory has not yet been applied. Can you find an area without applications of graph theory?

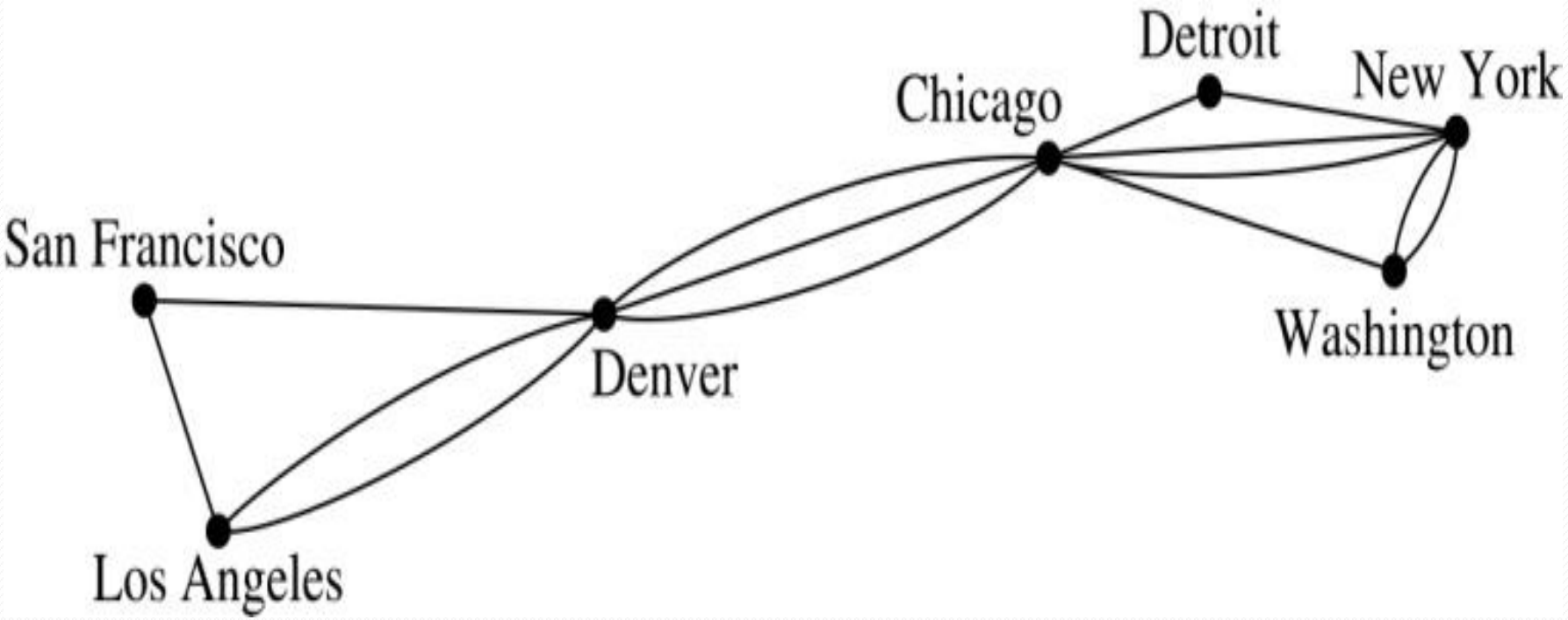
Graph Models: Computer Networks

- When we build a graph model, we use the appropriate type of graph to capture the important features of the application.
- We illustrate this process using graph models of different types of computer networks. In all these graph models, the vertices represent data centers and the edges represent communication links.
- To model a computer network where we are only concerned whether two data centers are connected by a communications link, we use a simple graph. This is the appropriate type of graph when we only care whether two data centers are directly linked (and not how many links there may be) and all communications links work in both directions.



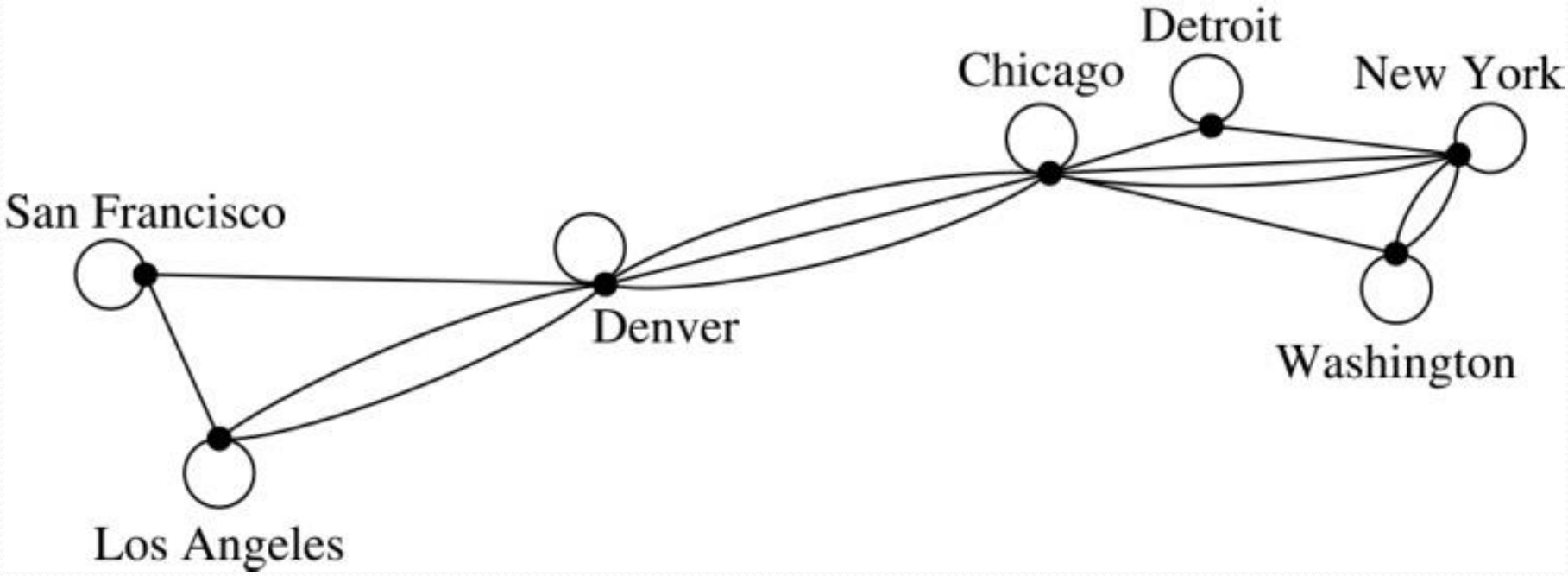
Graph Models: Computer Networks (*continued*)

- To model a computer network where we care about the number of links between data centers, we use a multigraph.



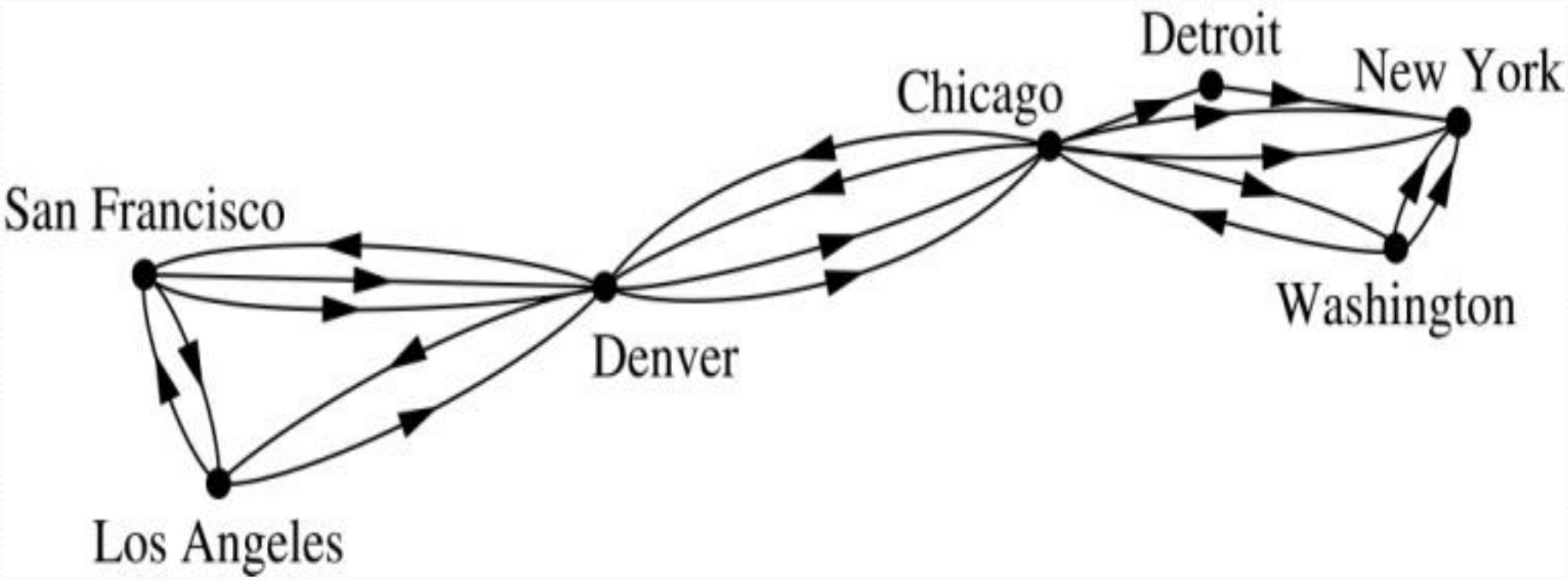
Graph Models: Computer Networks

- To model a computer network with diagnostic links at data centers, we use a pseudograph, as loops are needed.



Graph Models: Computer Networks

- To model a network with multiple one-way links, we use a directed multigraph. Note that we could use a directed graph without multiple edges if we only care whether there is at least one link from a data center to another data center.

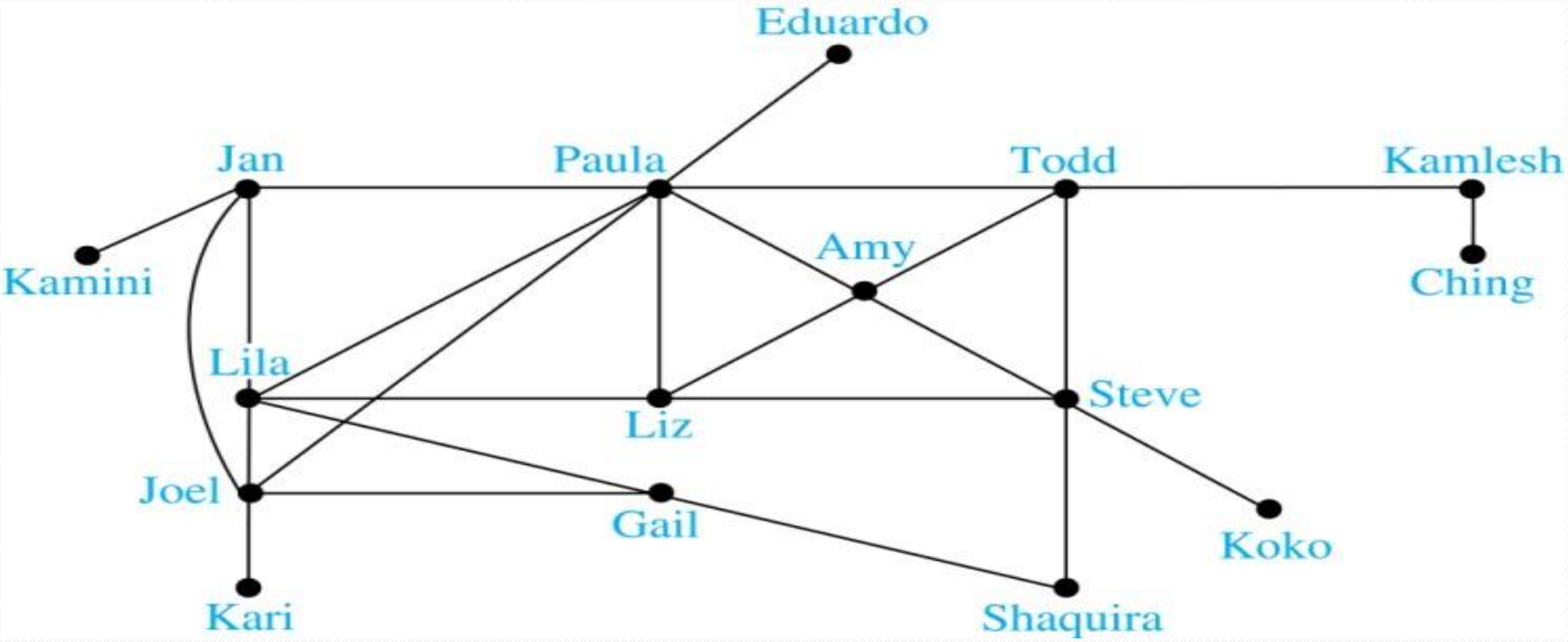


Graph Models: Social Networks

- Graphs can be used to model social structures based on different kinds of relationships between people or groups.
- In a *social network*, vertices represent individuals or organizations and edges represent relationships between them.
- Useful graph models of social networks include:
  - *friendship graphs* - undirected graphs where two people are connected if they are friends (in the real world, on Facebook, or in a particular virtual world, and so on.)
  - *collaboration graphs* - undirected graphs where two people are connected if they collaborate in a specific way
  - *influence graphs* - directed graphs where there is an edge from one person to another if the first person can influence the second person

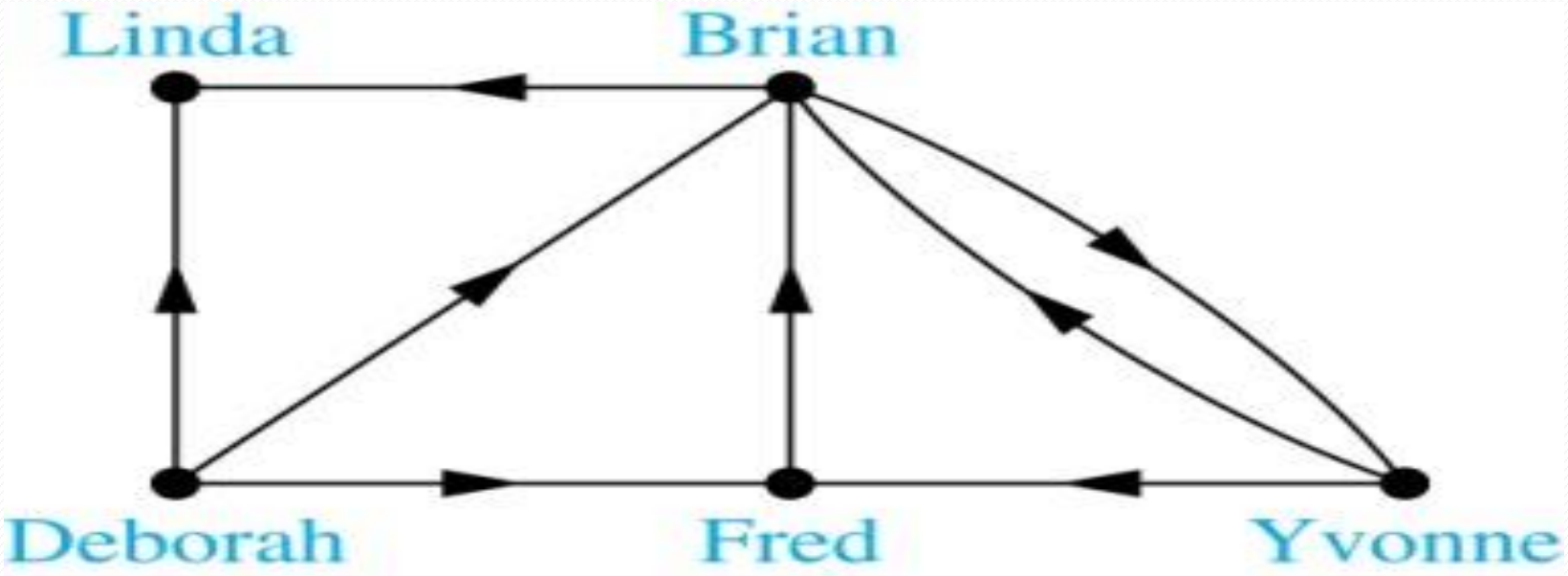
Graph Models: Social Networks

Example: A friendship graph where two people are connected if they are Facebook friends.



Graph Models: Social Networks

Example: An influence graph



Applications to Information Networks

- Graphs can be used to model different types of networks that link different types of information.
- In a *web graph*, web pages are represented by vertices and links are represented by directed edges.
  - A web graph models the web at a particular time.
  - We will explain how the web graph is used by search engines in Section 11.4.
- In a *citation network*:
  - Research papers in a particular discipline are represented by vertices.
  - When a paper cites a second paper as a reference, there is an edge from the vertex representing this paper to the vertex representing the second paper.

Transportation Graphs

- Graph models are extensively used in the study of transportation networks.
- Airline networks can be modeled using directed multigraphs where
  - airports are represented by vertices
  - each flight is represented by a directed edge from the vertex representing the departure airport to the vertex representing the destination airport
- Road networks can be modeled using graphs where
  - vertices represent intersections and edges represent roads.
  - undirected edges represent two-way roads and directed edges represent one-way roads.

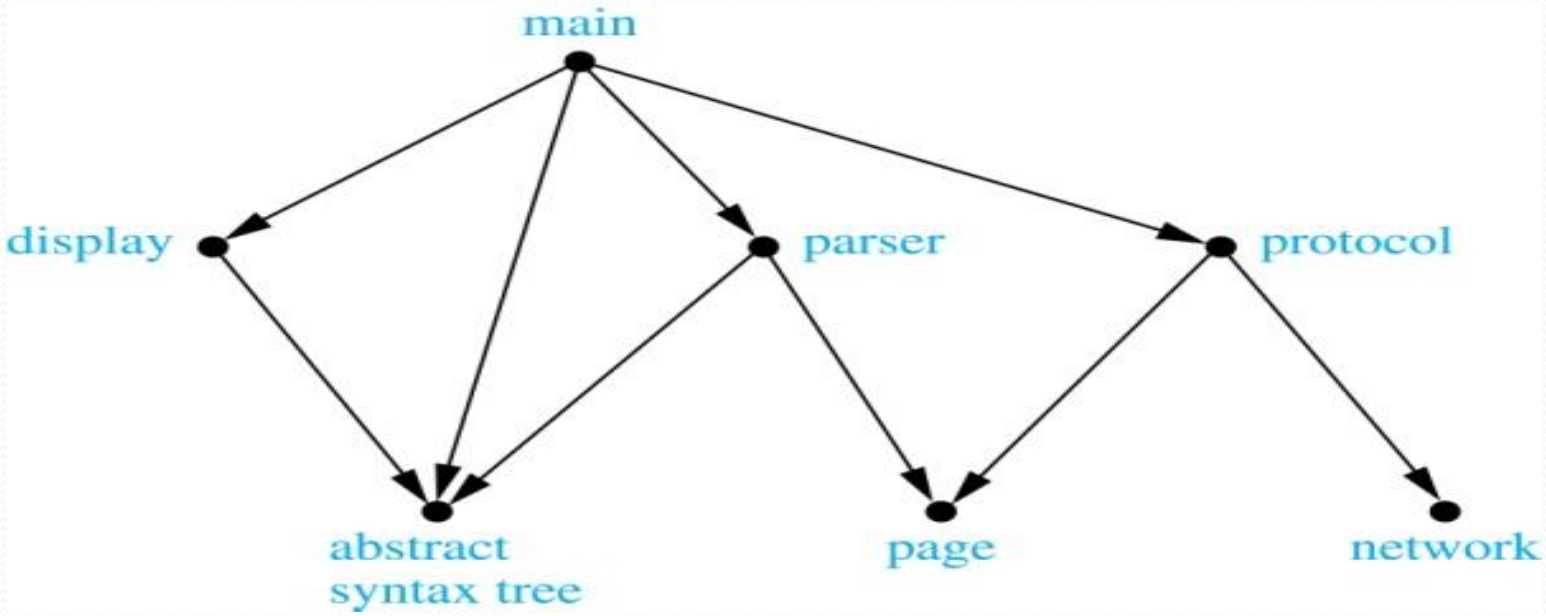
Software Design Applications

- Graph models are extensively used in software design. We will introduce two such models here; one representing the dependency between the modules of a software application and the other representing restrictions in the execution of statements in computer programs.
- When a top-down approach is used to design software, the system is divided into modules, each performing a specific task.
- We use a *module dependency graph* to represent the dependency between these modules. These dependencies need to be understood before coding can be done.

Software Design Applications

- In a module dependency graph vertices represent software modules and there is an edge from one module to another if the second module depends on the first.

Example: The dependencies between the seven modules in the design of a web browser are represented by this module dependency graph.



Software Design Applications

- We can use a directed graph called a *precedence graph* to represent which statements must have already been executed before we execute each statement.
  - Vertices represent statements in a computer program
  - There is a directed edge from a vertex to a second vertex if the second vertex cannot be executed before the first

Example: This precedence graph shows which statements must already have been executed before we can execute each of the six statements in the program.

S_1 $a := 0$

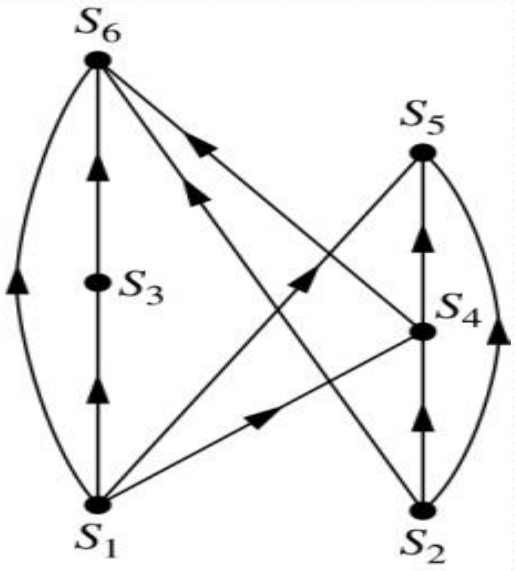
S_2 $b := 1$

S_3 $c := a + 1$

S_4 $d := b + a$

S_5 $e := d + 1$

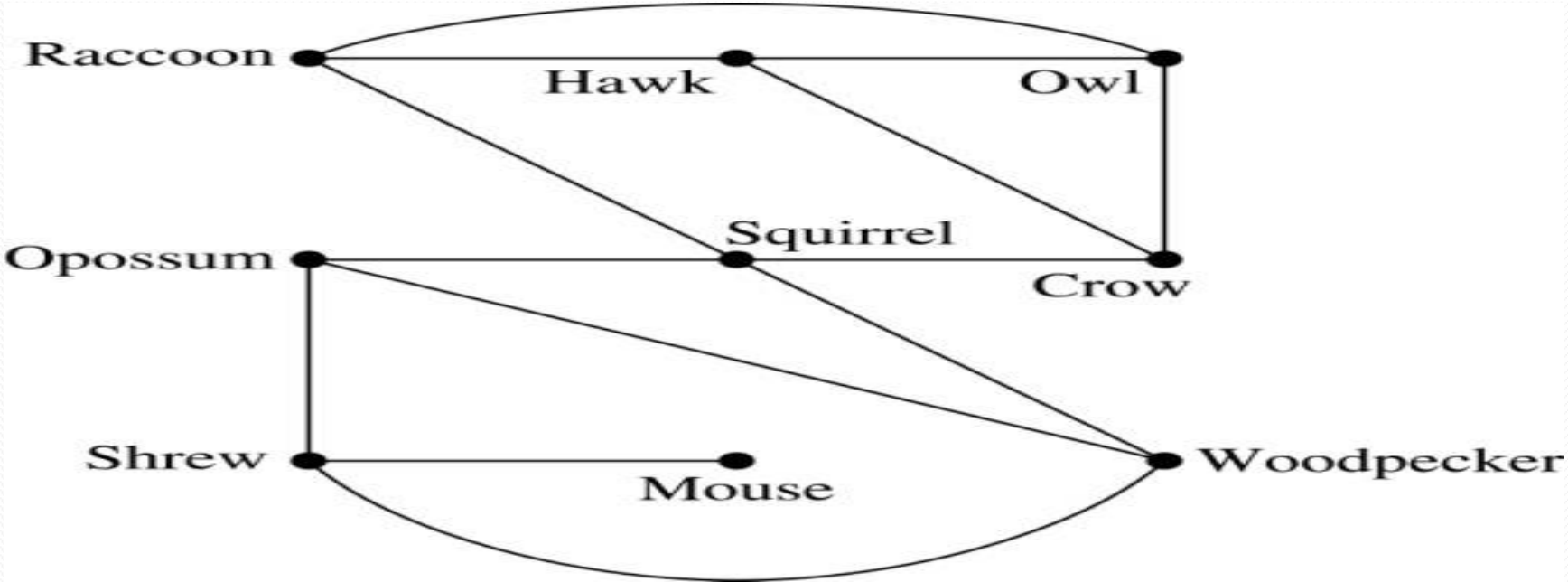
S_6 $e := c + d$



Biological Applications

- Graph models are used extensively in many areas of the biological science. We will describe two such models, one to ecology and the other to molecular biology.
- *Niche overlap graphs* model competition between species in an ecosystem
  - Vertices represent species and an edge connects two vertices when they represent species who compete for food resources.

Example: This is the niche overlap graph for a forest ecosystem with nine species.

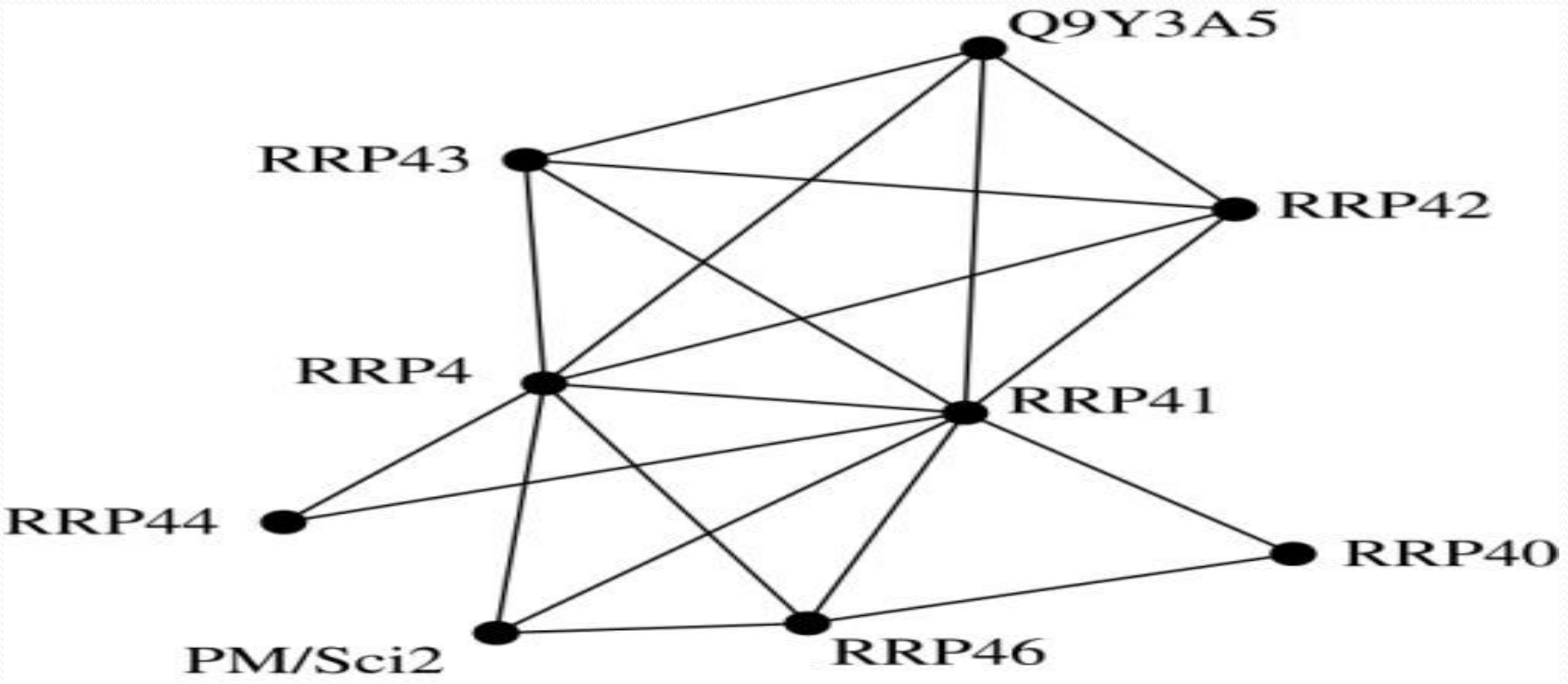


Biological Applications

- We can model the interaction of proteins in a cell using a *protein interaction network*.
- In a *protein interaction graph*, vertices represent proteins and vertices are connected by an edge if the proteins they represent interact.
- Protein interaction graphs can be huge and can contain more than 100,000 vertices, each representing a different protein, and more than 1,000,000 edges, each representing an interaction between proteins
- Protein interaction graphs are often split into smaller graphs, called *modules*, which represent the interactions between proteins involved in a particular function.

Biological Applications

Example: This is a module of the protein interaction graph of proteins that degrade RNA in a human cell.



Graph Terminology and Special Types of Graphs

Section 10.2

Section Summary

- Basic Terminology
- Some Special Types of Graphs
- Bipartite Graphs
- Bipartite Graphs and Matchings
- Some Applications of Special Types of Graphs
- New Graphs from Old

Basic Terminology

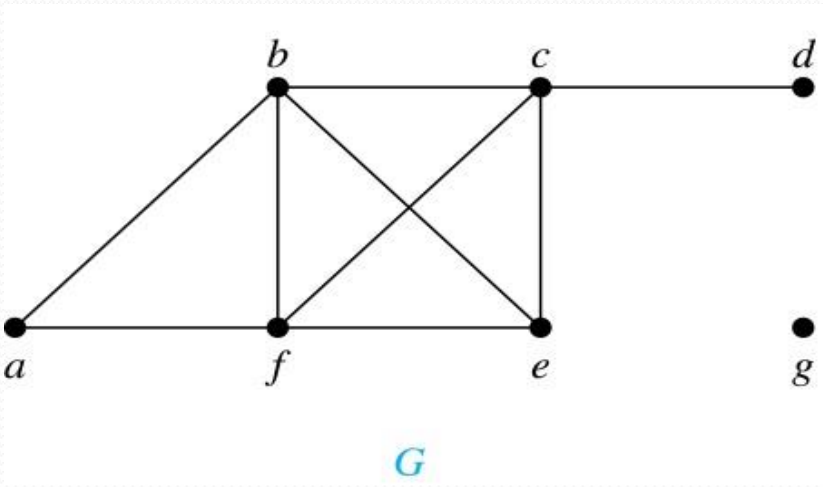
Definition 1. Two vertices *u*, *v* in an undirected graph *G* are called *adjacent* (or *neighbors*) in *G* if there is an edge *e* between *u* and *v*. Such an edge *e* is called *incident with* the vertices *u* and *v* and *e* is said to *connect u* and *v*.

Definition 2. 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, $N(A) = \bigcup_{v \in A} N(v)$.

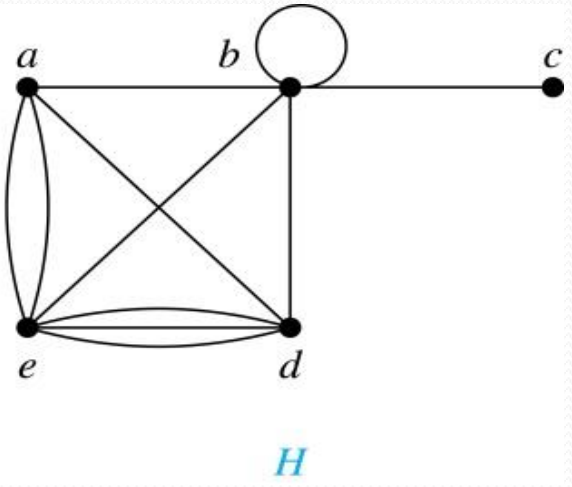
Definition 3. The *degree of a vertex in a undirected graph* is the number of edges incident with it, except that a loop at a vertex contributes two to the degree of that vertex. The degree of the vertex *v* is denoted by $\deg(v)$.

Degrees and Neighborhoods of Vertices

Example: What are the degrees and neighborhoods of the vertices in the graphs G and H?



G



H

Solution:

G: $\deg(a) = 2, \deg(b) = \deg(c) = \deg(f) = 4, \deg(d) = 1, \deg(e) = 3, \deg(g) = 0.$

$N(a) = \{b, f\}, N(b) = \{a, c, e, f\}, N(c) = \{b, d, e, f\}, N(d) = \{c\},$

$N(e) = \{b, c, f\}, N(f) = \{a, b, c, e\}, N(g) = \varnothing$.

H: $\deg(a) = 4, \deg(b) = \deg(e) = 6, \deg(c) = 1, \deg(d) = 5.$

$N(a) = \{b, d, e\}, N(b) = \{a, b, c, d, e\}, N(c) = \{b\}, N(d) = \{a, b, e\},$

$N(e) = \{a, b, d\}.$

Handshaking Theorem

Theorem: If G is any graph, then the sum of the degrees of all the vertices of G equals twice the number of edges of G.

Specifically, if the vertices of G are **v1**, **v2**, …, **vn**, where n is a positive integer, then

the total degree of G = deg(v1) + deg(v2) + … + deg(vn)
= 2 . (the number of edges of G)

PROOF:

- Each edge "e" of G connects its end points vi and vj. This edge, therefore contributes 1 to the degree of vi and 1 to the degree of vj.
- If "e" is a loop, then it is counted twice in computing the degree of the vertex on which it is incident.
- Accordingly, each edge of G contributes 2 to the total degree of G.

Thus, the total degree of G = 2. (the number of edges of G)

COROLLARY: The total degree of G is an even number

Degree of Vertices

Theorem: An undirected graph has an even number of vertices of odd degree.

***Proof*:** Let V_1 be the vertices of even degree and V_2 be the vertices of odd degree in an undirected graph $G = (V, E)$ with m edges. Then

even → $$2m = \sum_{v \in V} \deg(v) = \sum_{v \in V_1} \deg(v) + \sum_{v \in V_2} \deg(v).$$

must be even since deg(v) is even for each $v \in V_1$

This sum must be even because $2m$ is even and the sum of the degrees of the vertices of even degrees is also even. Because this is the sum of the degrees of all vertices of odd degree in the graph, there must be an even number of such vertices.

Handshaking Theorem

We now give two examples illustrating the usefulness of the handshaking theorem.

Example: How many edges are there in a graph with 10 vertices of degree six?

Solution: Because the sum of the degrees of the vertices is $6 \cdot 10 = 60$, the handshaking theorem tells us that $2m = 60$. So the number of edges $m = 30$.

Example: If a graph has 5 vertices, can each vertex have degree 3?

Solution: This is not possible by the handshaking thoerem, because the sum of the degrees of the vertices $3 \cdot 5 = 15$ is odd.

Directed Graphs

Recall the definition of a directed graph.

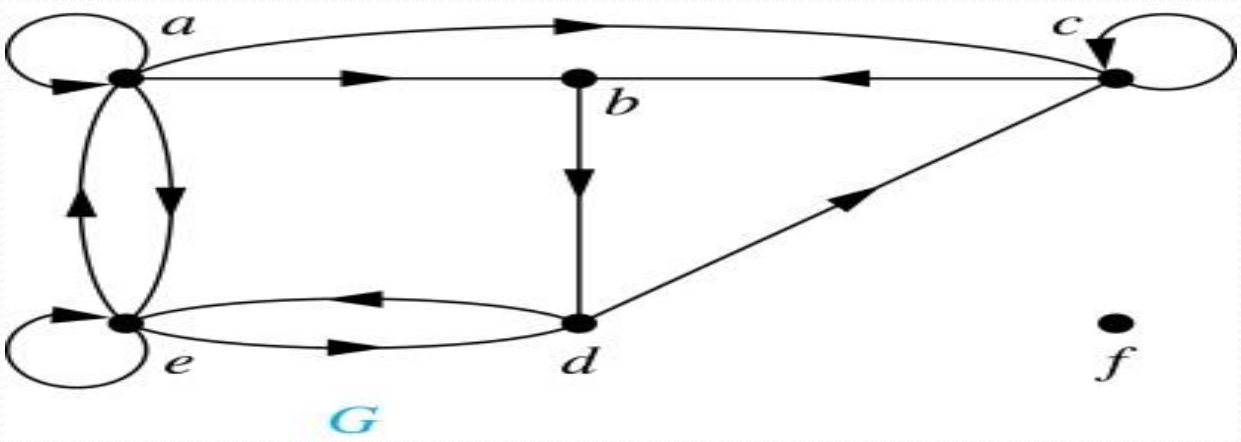
Definition: An *directed graph* $G = (V, E)$ consists of V, a nonempty set of *vertices* (or *nodes*), and E, a set of *directed edges* or *arcs*. Each edge is an ordered pair of vertices. The directed edge (u,v) is said to start at u and end at v.

Definition: Let (u,v) be an edge in G. Then u is the *initial vertex* of this edge and is *adjacent to* v and v is the *terminal* (or *end*) *vertex* of this edge and is *adjacent from* u. The initial and terminal vertices of a loop are the same.

Directed Graphs (*continued*)

Definition: The *in-degree of a vertex v*, denoted $deg^-(v)$, is the number of edges which terminate at v. The *out-degree of v*, denoted $deg^+(v)$, is the number of edges with v as their initial vertex. Note that a loop at a vertex contributes 1 to both the in-degree and the out-degree of the vertex.

Example: In the graph G we have



$\deg^-(a) = 2, \deg^-(b) = 2, \deg^-(c) = 3, \deg^-(d) = 2, \deg^-(e) = 3, \deg^-(f) = 0.$

$\deg^+(a) = 4, \deg^+(b) = 1, \deg^+(c) = 2, \deg^+(d) = 2, \deg^+(e) = 3, \deg^+(f) = 0.$

Directed Graphs (*continued*)

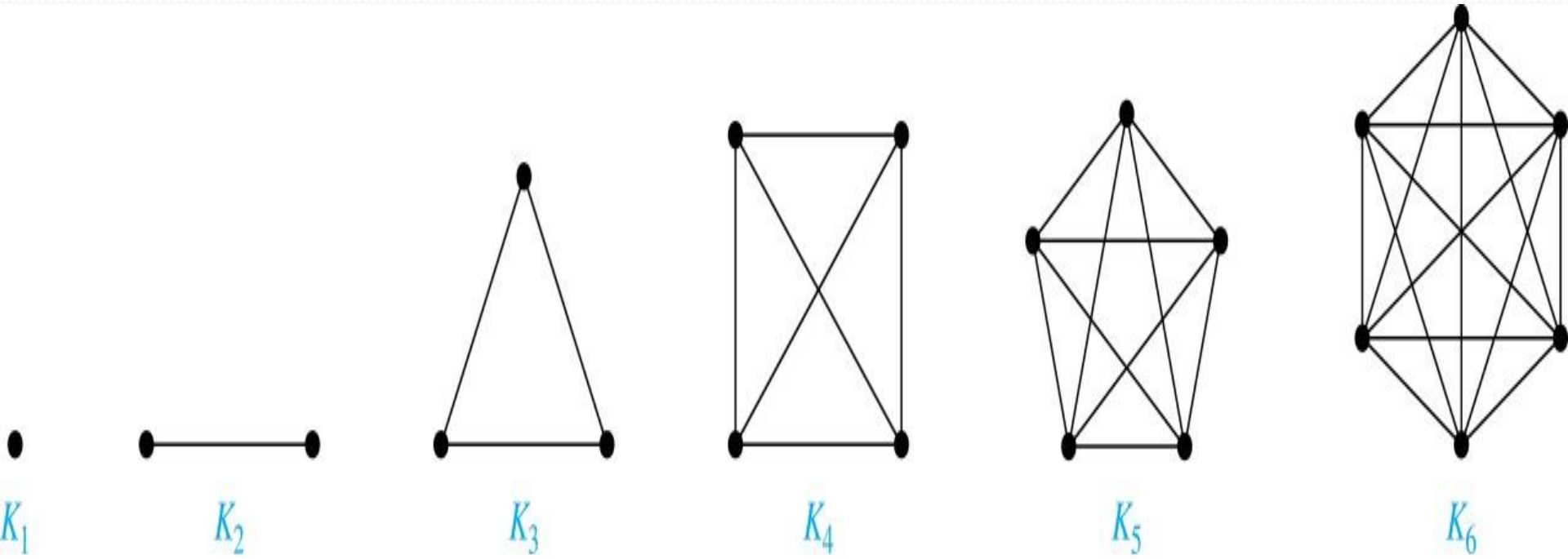
Theorem 3: Let $G = (V, E)$ be a graph with directed edges. Then:

$$|E| = \sum_{v \in V} deg^-(v) = \sum_{v \in V} deg^+(v).$$

Proof: The first sum counts the number of outgoing edges over all vertices and the second sum counts the number of incoming edges over all vertices. It follows that both sums equal the number of edges in the graph.

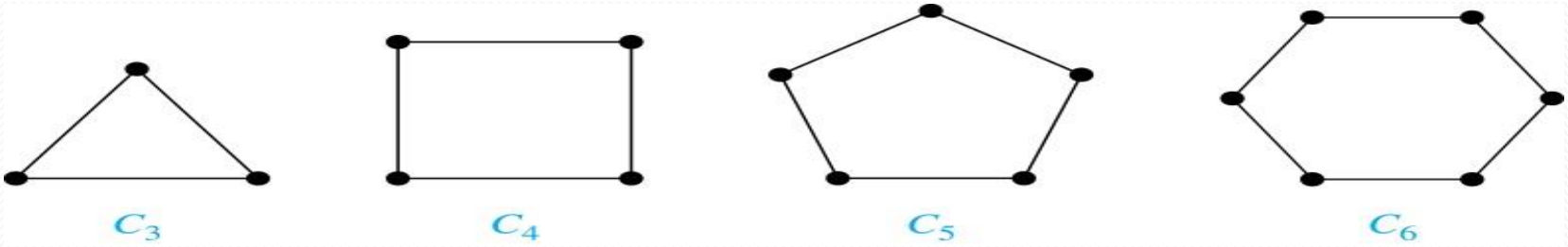
Special Types of Simple Graphs: Complete Graphs

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

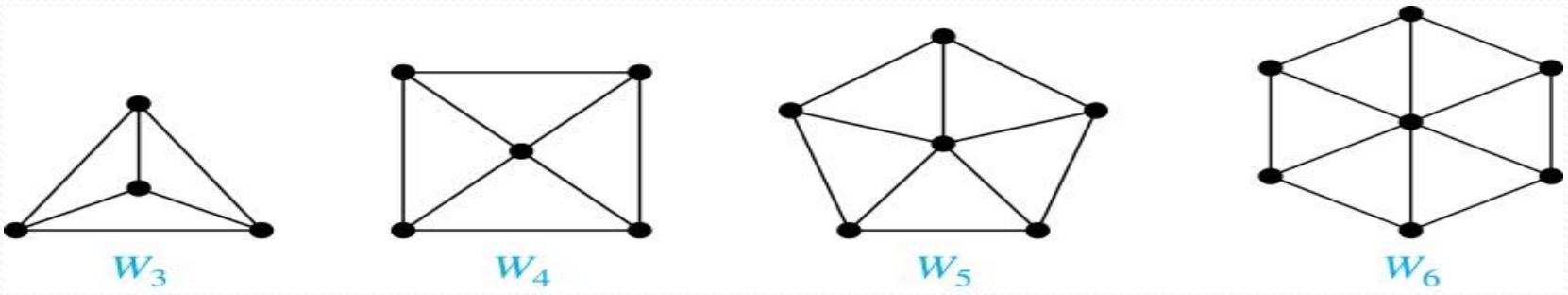


Special Types of Simple Graphs: Cycles and Wheels

A *cycle* C_n for $n \geq 3$ consists of n vertices $v_1, v_2, \cdots, v_n$, and edges $\{v_1, v_2\}, \{v_2, v_3\}, \cdots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$.

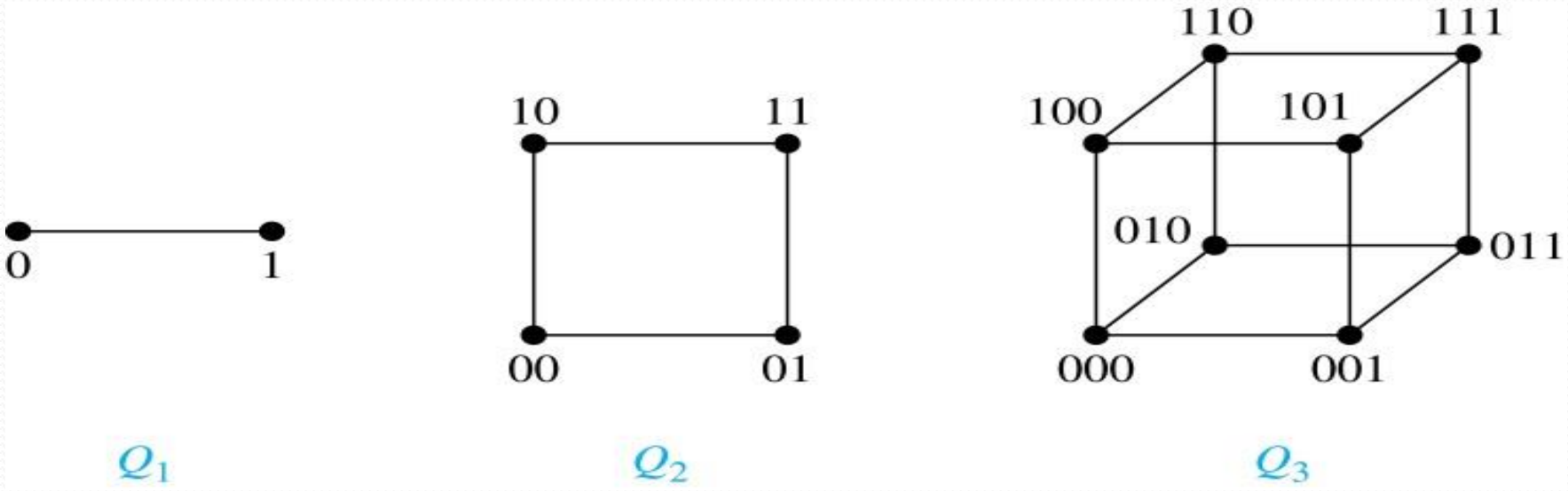


A *wheel* W_n is obtained by adding an additional vertex to a cycle C_n for $n \geq 3$ and connecting this new vertex to each of the n vertices in C_n by new edges.



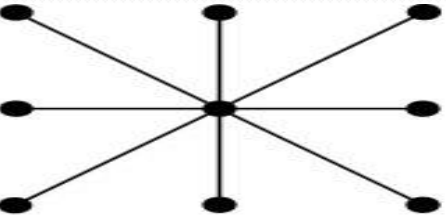
Special Types of Simple Graphs: *n*-Cubes

An *n-dimensional hypercube*, or *n-cube*, Q_n, is a graph with 2^n vertices representing all bit strings of length n, where there is an edge between two vertices that differ in exactly one bit position.

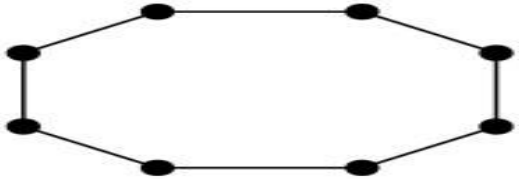


Special Types of Graphs and Computer Network Architecture

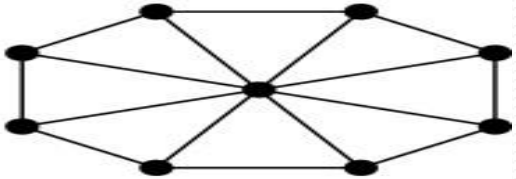
Various special graphs play an important role in the design of computer networks.



(a)



(b)

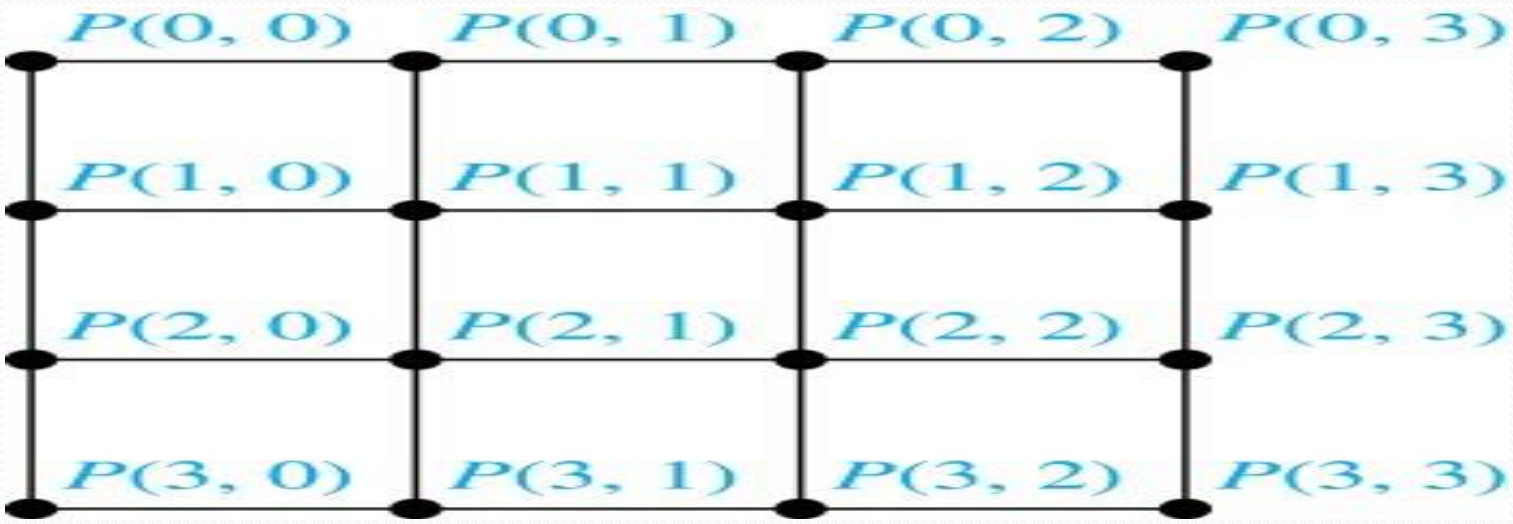


(c)

- Some local area networks use a *star topology*, which is a complete bipartite graph $K_{1,n}$, as shown in (a). All devices are connected to a central control device.
- Other local networks are based on a *ring topology*, where each device is connected to exactly two others using C_n, as illustrated in (b). Messages may be sent around the ring.
- Others, as illustrated in (c), use a W_n – based topology, combining the features of a star topology and a ring topology.

Special Types of Graphs and Computer Network Architecture

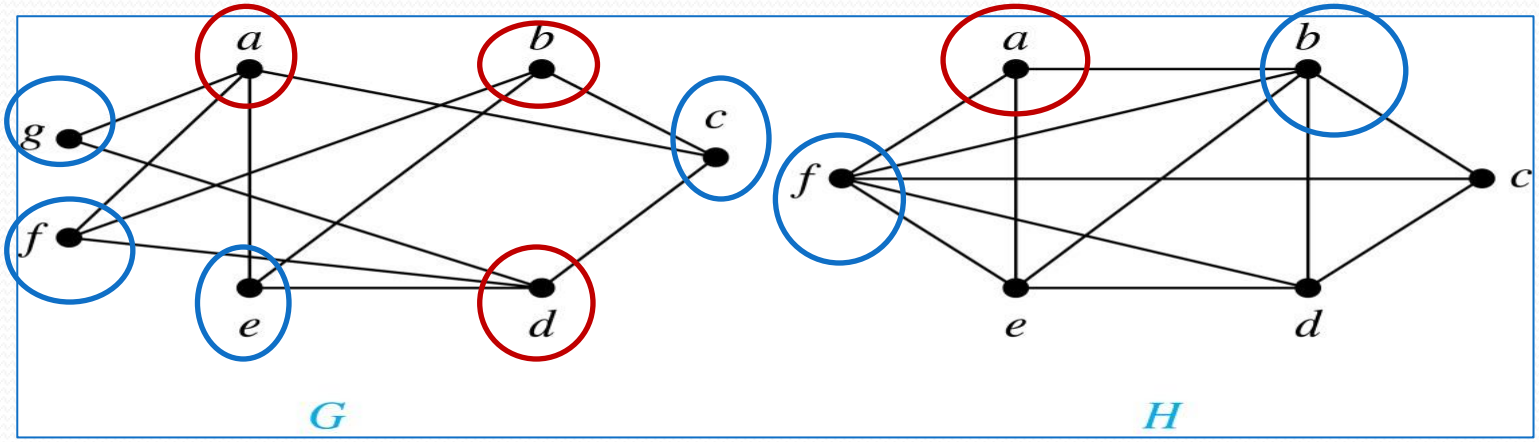
- Various special graphs also play a role in parallel processing where processors need to be interconnected as one processor may need the output generated by another.
- The n-dimensional hypercube, or n-cube, Qn, is a common way to connect processors in parallel, e.g., Intel Hypercube.
- Another common method is the mesh network, illustrated here for 16 processors.



Bipartite Graphs

Definition: A simple graph G is bipartite if V can be partitioned into two disjoint subsets V_1 and V_2 such that every edge connects a vertex in V_1 and a vertex in V_2. In other words, there are no edges which connect two vertices in V_1 or in V_2.

- It is not hard to show that an equivalent definition of a bipartite graph is a graph where it is possible to color the vertices red or blue so that no two adjacent vertices are the same color.



G is bipartite

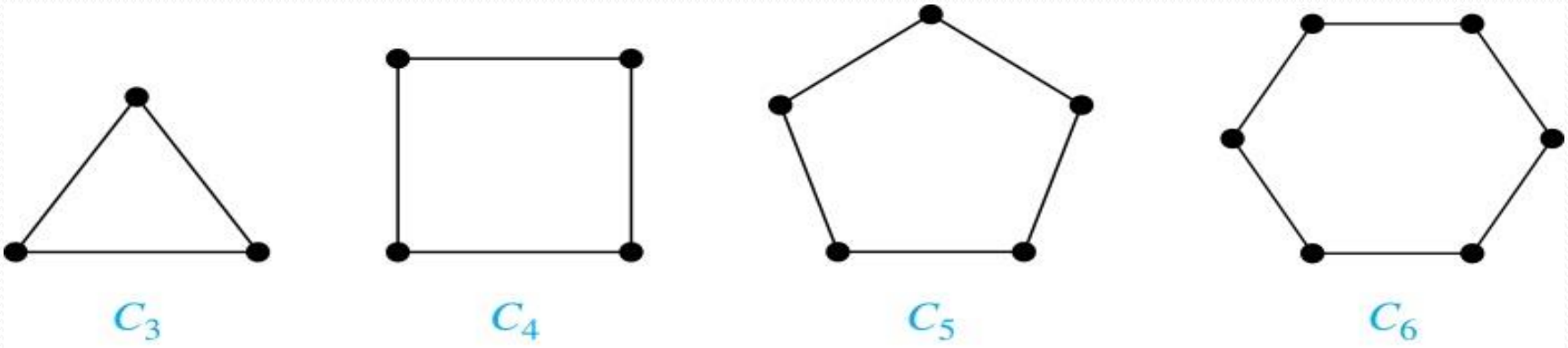
H is not bipartite since if we color a red, then the adjacent vertices f and b must both be blue.

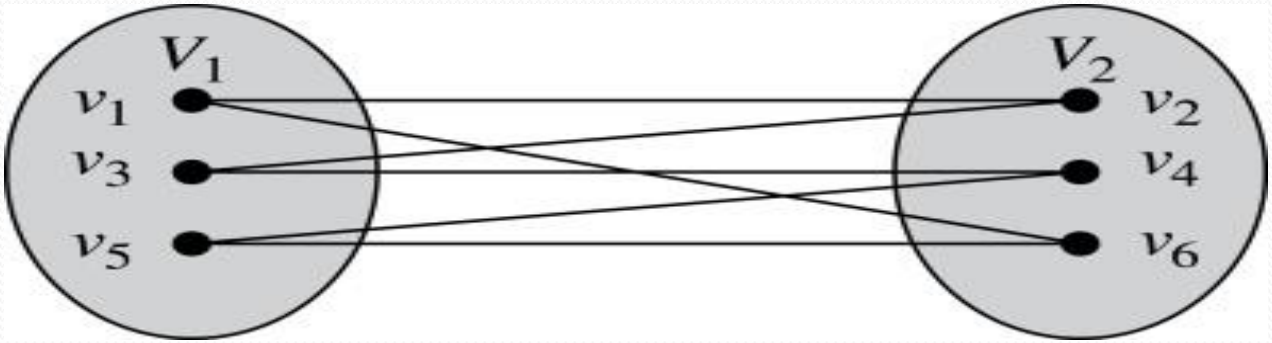
Bipartite Graphs (*continued*)

Example: Show that C_6 is bipartite.

Solution: We can partition the vertex set into

$V_1 = \{v_1, v_3, v_5\}$ and $V_2 = \{v_2, v_4, v_6\}$ so that every edge of C_6 connects a vertex in V_1 and V_2 .

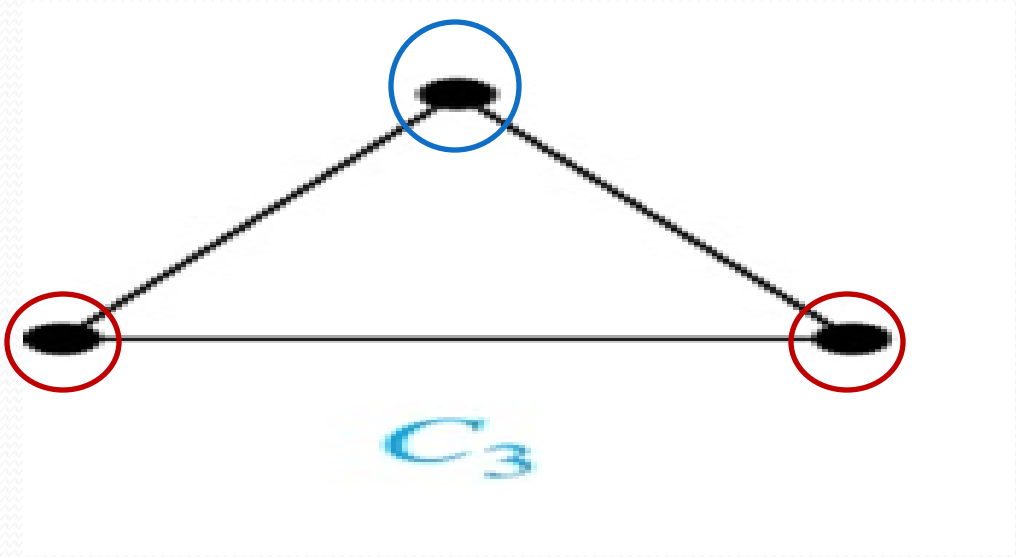




Bipartite Graphs (*continued*)

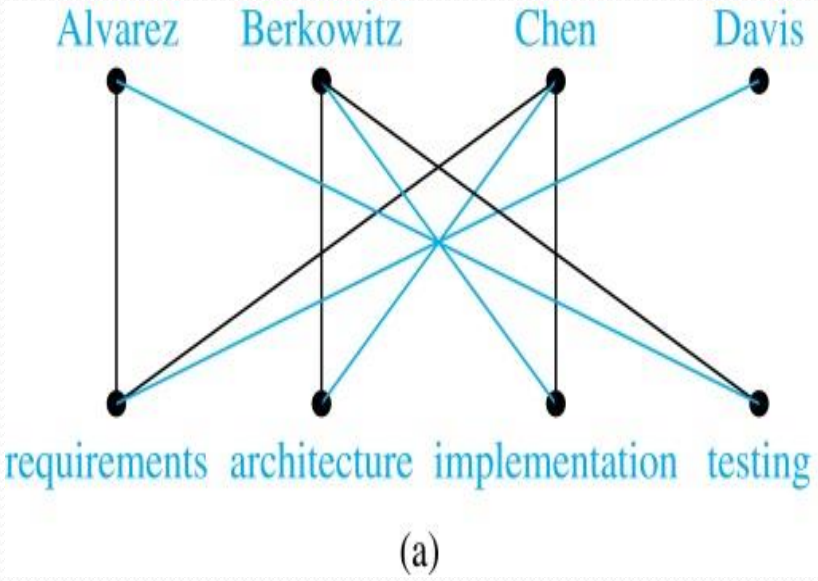
Example: Show that C_3 is not bipartite.

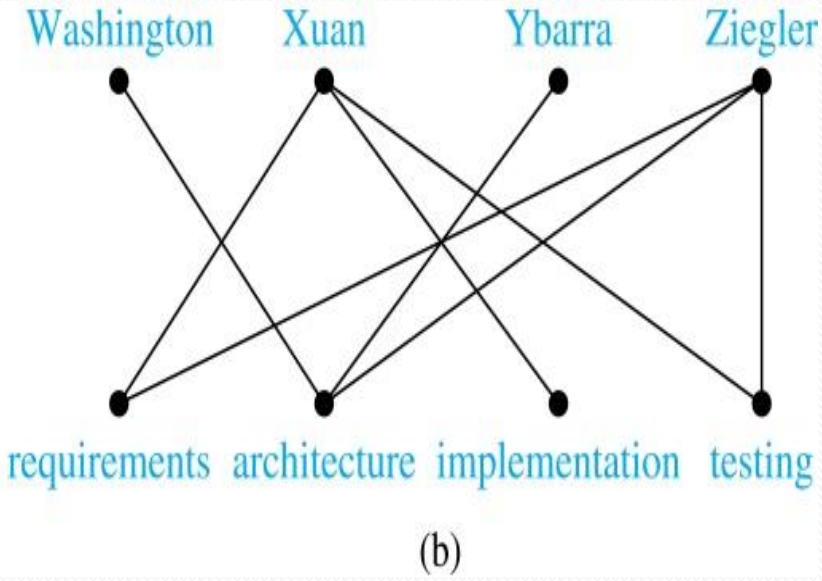
Solution: If we divide the vertex set of C_3 into two nonempty sets, one of the two must contain two vertices. But in C_3 every vertex is connected to every other vertex. Therefore, the two vertices in the same partition are connected. Hence, C_3 is not bipartite.



Bipartite Graphs and Matchings

- Bipartite graphs are used to model applications that involve matching the elements of one set to elements in another, for example:
- *Job assignments* - vertices represent the jobs and the employees, edges link employees with those jobs they have been trained to do. A common goal is to match jobs to employees so that the most jobs are done.

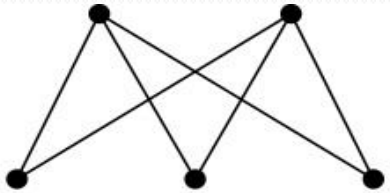




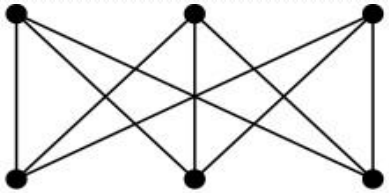
Complete Bipartite Graphs

Definition: A *complete bipartite graph* $K_{m,n}$ is a graph that has its vertex set partitioned into two subsets V_1 of size m and V_2 of size n such that there is an edge from every vertex in V_1 to every vertex in V_2.

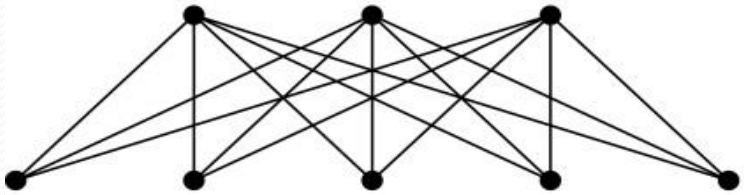
Example: We display four complete bipartite graphs here.



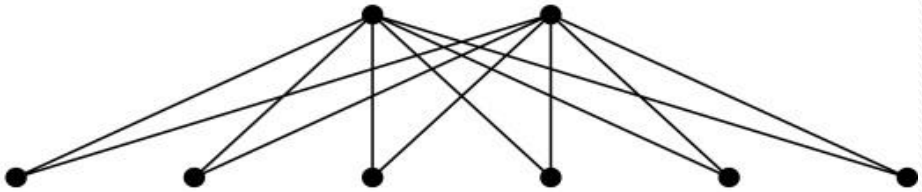
$K_{2,3}$



$K_{3,3}$



$K_{3,5}$

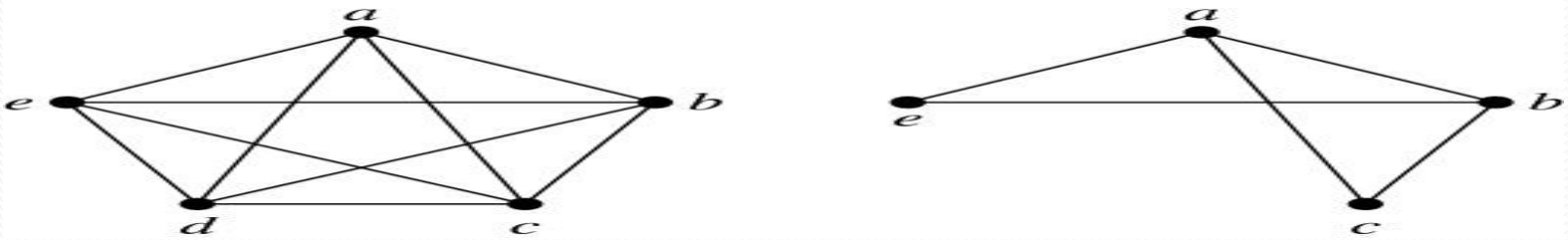


$K_{2,6}$

New Graphs from Old

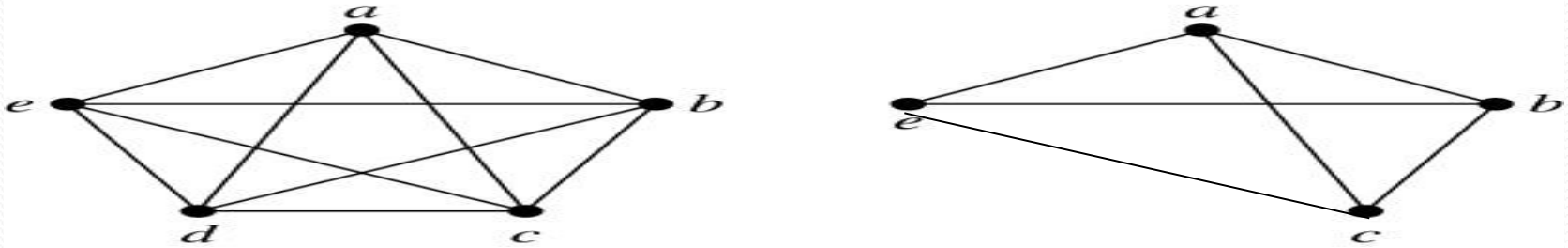
Definition: A *subgraph of a graph* $G = (V,E)$ is a graph (W,F), where $W \subset V$ and $F \subset E$. A subgraph H of G is a proper subgraph of G if $H \neq G$.

Example: Here we show K_5 and one of its subgraphs.



Definition: Let $G = (V, E)$ be a simple graph. 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 are in W.

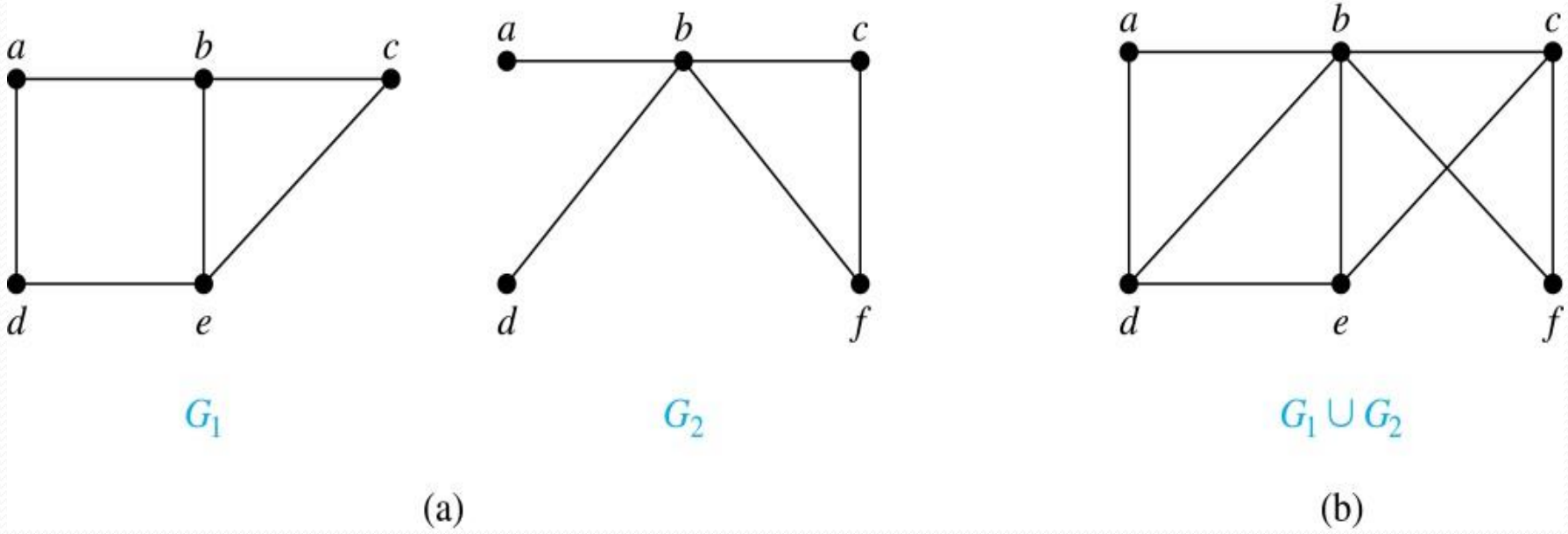
Example: Here we show K_5 and the subgraph induced by $W = \{a,b,c,e\}$.



New Graphs from Old (*continued*)

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 $G_1 \cup G_2$.

Example:



Representing Graphs and Graph Isomorphism

Section 10.3

Section Summary

- Adjacency Lists
- Adjacency Matrices
- Incidence Matrices
- Isomorphism of Graphs

Representing Graphs: Adjacency Lists

Definition: An *adjacency list* can be used to represent a graph with no multiple edges by specifying the vertices that are adjacent to each vertex of the graph.

Example:

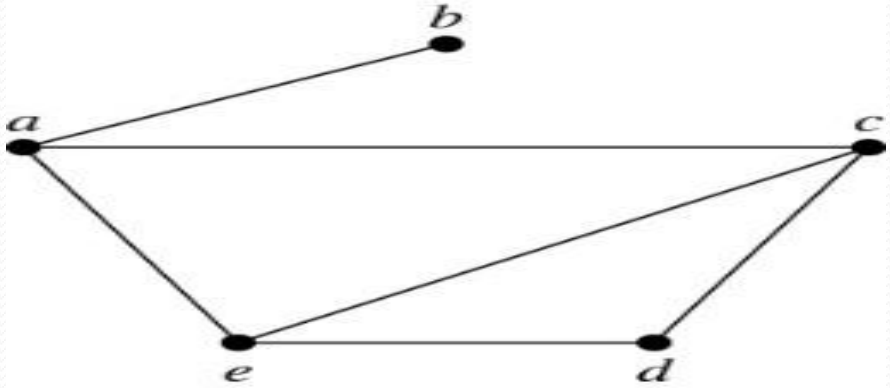


TABLE 1 An Adjacency List for a Simple Graph.

Vertex	*Adjacent Vertices*
a	b, c, e
b	a
c	a, d, e
d	c, e
e	a, c, d

Representing Graphs: Adjacency Lists

Example:

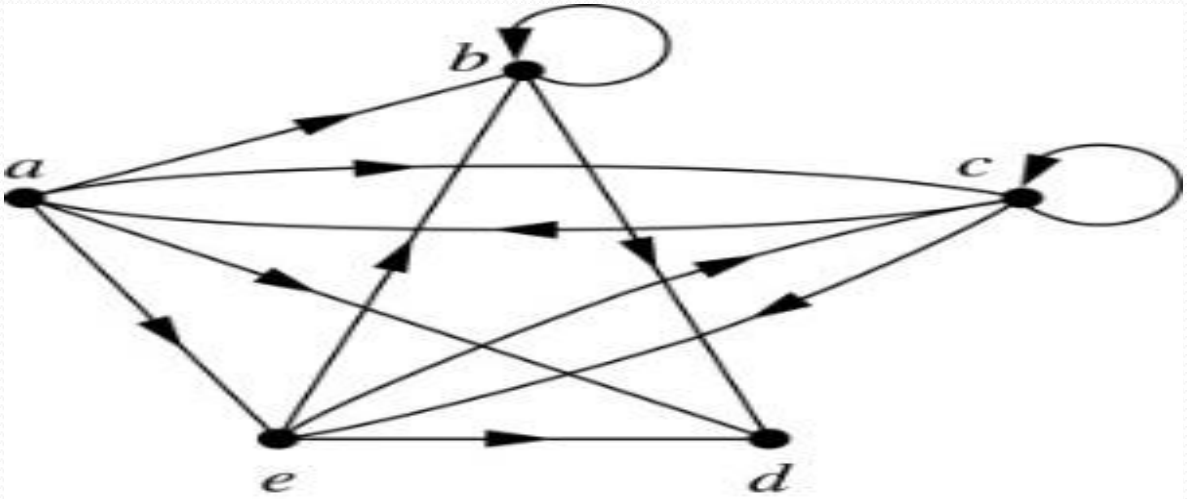


TABLE 2 An Adjacency List for a Directed Graph.

Initial Vertex	*Terminal Vertices*
a	b, c, d, e
b	b, d
c	a, c, e
d	
e	b, c, d

Representation of Graphs: Adjacency Matrices

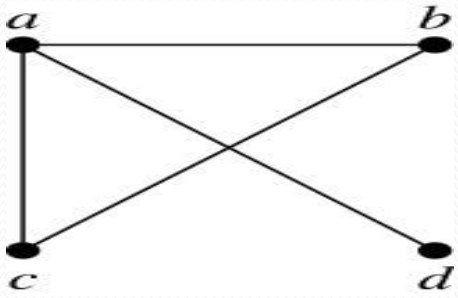
Definition: Suppose that $G = (V, E)$ is a simple graph where $|V| = n$. Arbitrarily list the vertices of G as $v_1, v_2, \ldots, v_n$. The *adjacency matrix* $\mathbf{A}_G$ of G, with respect to the listing of vertices, is the $n \times n$ zero-one matrix with 1 as its (i, j)th entry when v_i and v_j are adjacent, and 0 as its (i, j)th entry when they are not adjacent.

- In other words, if the graphs adjacency matrix is $\mathbf{A}_G = [a_{ij}]$, then

$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of } G, \\ 0 & \text{otherwise.} \end{cases}$$

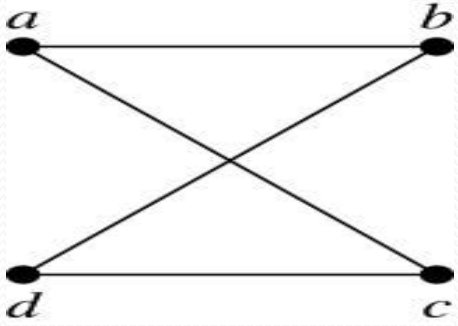
Adjacency Matrices (*continued*)

Example:



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

The ordering of vertices is a, b, c, d.



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

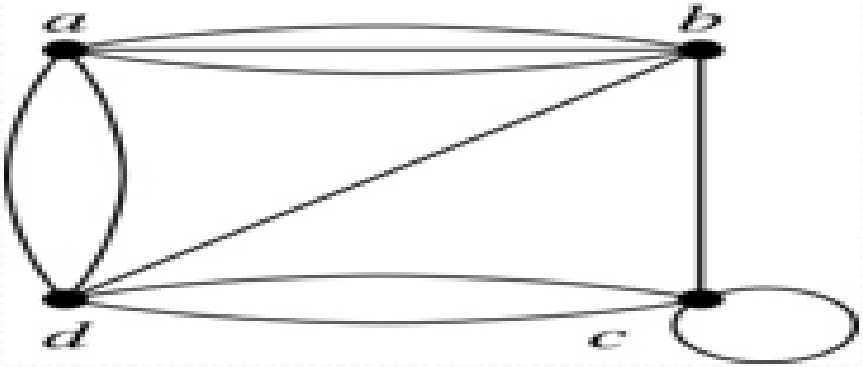
The ordering of vertices is a, b, c, d.

When a graph is sparse, that is, it has few edges relatively to the total number of possible edges, it is much more efficient to represent the graph using an adjacency list than an adjacency matrix. But for a dense graph, which includes a high percentage of possible edges, an adjacency matrix is preferable.

Note: The adjacency matrix of a simple graph is symmetric, i.e., $a_{ij} = a_{ji}$
Also, since there are no loops, each diagonal entry a_{ij} for $i = 1, 2, 3, \ldots, n$, is 0.

Adjacency Matrices (*continued*)

- Adjacency matrices can also be used to represent graphs with loops and multiple edges.
- A loop at the vertex vi is represented by a 1 at the (i, j)th position of the matrix.
- When multiple edges connect the same pair of vertices vi and vj, (or if multiple loops are present at the same vertex), the (i, j)th entry equals the number of edges connecting the pair of vertices.
- Example: We give the adjacency matrix of the pseudograph shown here using the ordering of vertices a, b, c, d.



$$\begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}$$

Adjacency Matrices (*continued*)

- Adjacency matrices can also be used to represent directed graphs. The matrix for a directed graph $G = (V, E)$ has a 1 in its (i, j)th position if there is an edge from v_i to v_j, where $v_1, v_2, \ldots v_n$ is a list of the vertices.
  - In other words, if the graphs adjacency matrix is $\mathbf{A}_G = [a_{ij}]$, then

$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of } G, \\ 0 & \text{otherwise.} \end{cases}$$

  - The adjacency matrix for a directed graph does not have to be symmetric, because there may not be an edge from v_i to v_j, when there is an edge from v_j to v_i.
  - To represent directed multigraphs, the value of a_{ij} is the number of edges connecting v_i to v_j.

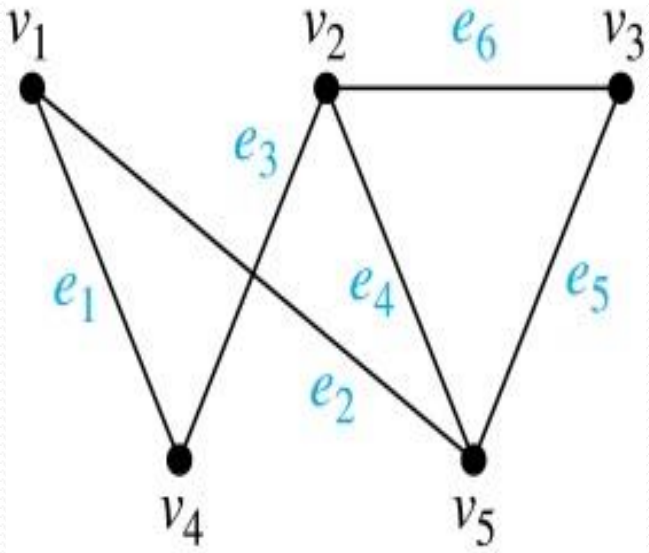
Representation of Graphs: Incidence Matrices

Definition: Let $G = (V, E)$ be an undirected graph with vertices where $v_1, v_2, \ldots v_n$ and edges $e_1, e_2, \ldots e_m$. The incidence matrix with respect to the ordering of V and E is the $n \times m$ matrix $\mathbf{M} = [m_{ij}]$, where

$$m_{ij} = \begin{cases} 1 & \text{when edge } e_j \text{ is incident with } v_i, \\ 0 & \text{otherwise.} \end{cases}$$

Incidence Matrices (*continued*)

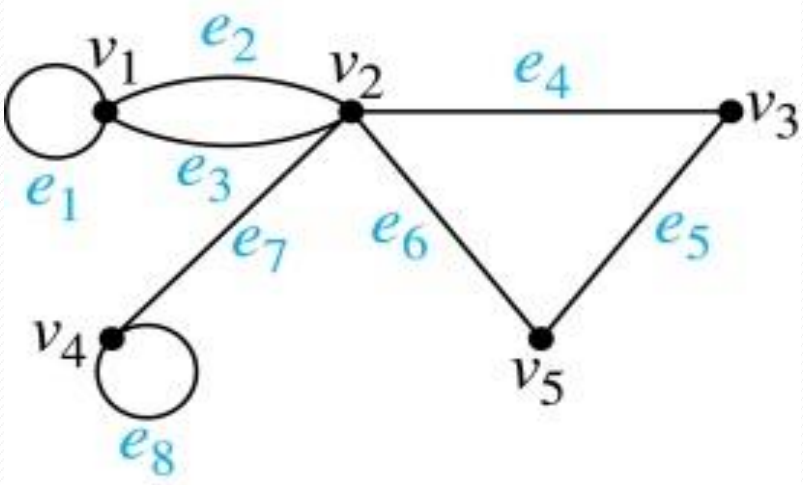
Example: Simple Graph and Incidence Matrix



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

The rows going from top to bottom represent v_1 through v_5 and the columns going from left to right represent e_1 through e_6.

Example: Pseudograph and Incidence Matrix



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

The rows going from top to bottom represent v_1 through v_5 and the columns going from left to right represent e_1 through e_8.

Isomorphism of Graphs

Definition: The simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are *isomorphic* if there is a one-to-one and onto function f from V_1 to V_2 with the property that a and b are adjacent in G_1 if and only if $f(a)$ and $f(b)$ are adjacent in G_2 , for all a and b in V_1 . Such a function f is called an *isomorphism*. Two simple graphs that are not isomorphic are called *nonisomorphic*.

Isomorphism of Graphs

- It is difficult to determine whether two simple graphs are isomorphic using brute force because there are $n!$ possible one-to-one correspondences between the vertex sets of two simple graphs with n vertices.
- The best algorithms for determining weather two graphs are isomorphic have exponential worst case complexity in terms of the number of vertices of the graphs.
- Sometimes it is not hard to show that two graphs are not isomorphic. We can do so by finding a property, preserved by isomorphism, that only one of the two graphs has. Such a property is called *graph invariant*.
- There are many different useful graph invariants that can be used to distinguish nonisomorphic graphs, such as the number of vertices, number of edges, and degree sequence (list of the degrees of the vertices in nonincreasing order). We will encounter others in later sections of this chapter.

ISOMORPHIC INVARIANT

- A property P is called an isomorphic invariant if, and only if, given any graphs G and G',
- If G has property P and G' is isomorphic to G, then G' has property P.

THEOREM OF ISOMORPHIC INVARIANT

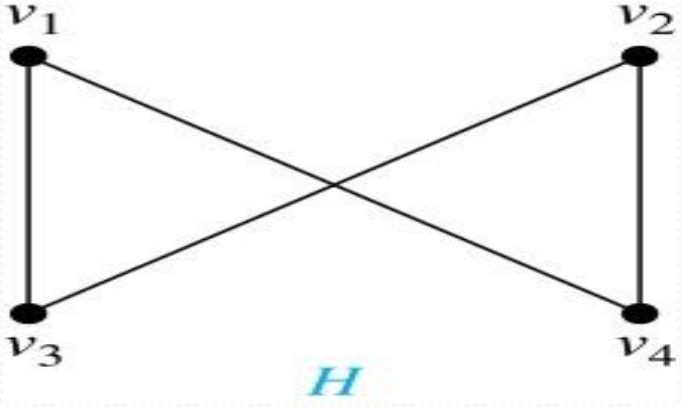
Each of the following properties is an invariant for graph isomorphism, where n, m and k are all non-negative integers, if the graph:

1. has n vertices.
2. has m edges.
3. has a vertex of degree k.
4. has m vertices of degree k.
5. has a circuit of length k.
6. has a simple circuit of length k.
7. has m simple circuits of length k.
8. is connected.
9. has an Euler circuit.
10. has a Hamiltonian circuit.

Isomorphism of Graphs (*cont.*)

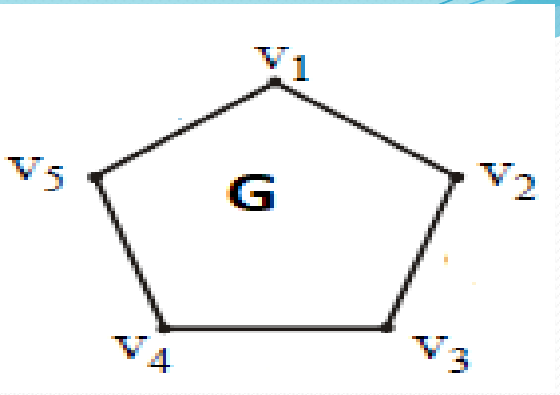
Example: Show that the graphs $G = (V, E)$ and $H = (W, F)$ are isomorphic.

Solution: The function f with $f(u_1) = v_1, f(u_2) = v_4, f(u_3) = v_3$, and $f(u_4)$ correspondence between V and W. Note that adjacent vertices in G are u_1 and u_2, u_1 and u_3, u_2 and u_4, and u_3 and u_4. Each of the pairs $f(u_1) = v_1$ and $f(u_2) = v_4$, $f(u_1) = v_1$ and $f(u_3) = v_3$, $f(u_2) = v_4$ and $f(u_4) = v_2$, and $f(u_3) = v_3$ and $f(u_4) = v_2$ consists of two adjacent vertices in H.



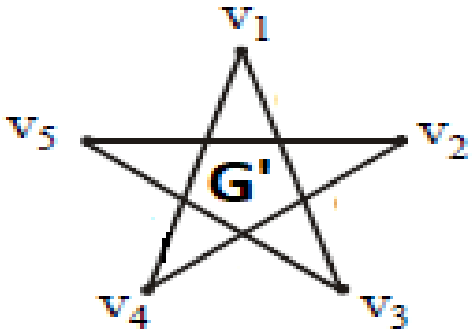
Isomorphism of Graphs

Example: Show that the graphs $G = (V, E)$ and $G' = (W, F)$ are isomorphic.



Solution: The function f with

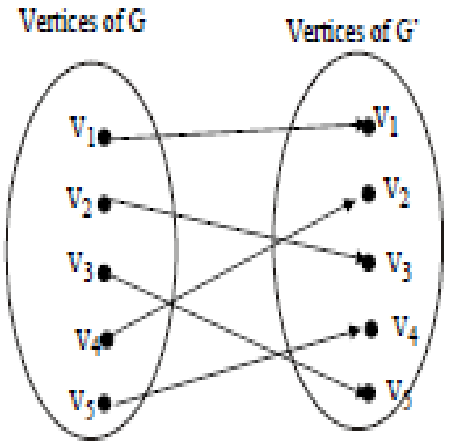
f(v1) = v1,f(v2) = v3, f(v3) = v5, f(v4) = v2 and f(v5) = v4 is a one-to-one correspondence between V and W.



Note that adjacent vertices in G are v1 and v2, v2 and v3, v3 and v4, v4 and v5 and v5 and v1.

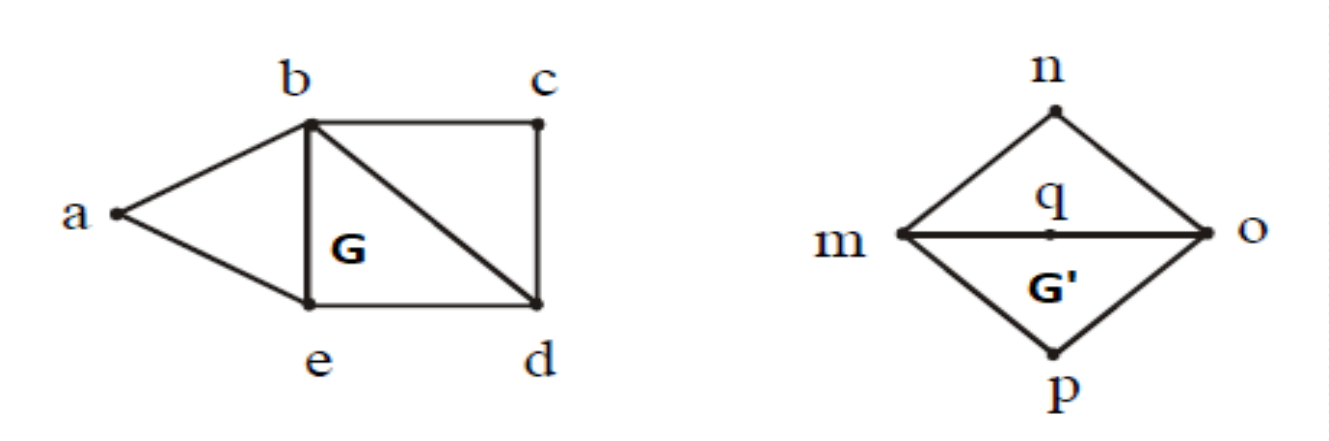
Each of the pairs f(v1) = v1 and f(v2) = v3, f(v2) = v3 and f(v3) = v5 , f(v3) = v5 and

f(v4) = v2 , f(v4) = v2 and f(v5) = v4 and f(v5) = v4 and f(v1) = v1 consists of two adjacent vertices in H.



Isomorphism of Graphs

EXAMPLE: Determine whether the graph G and G' given below are isomorphic.



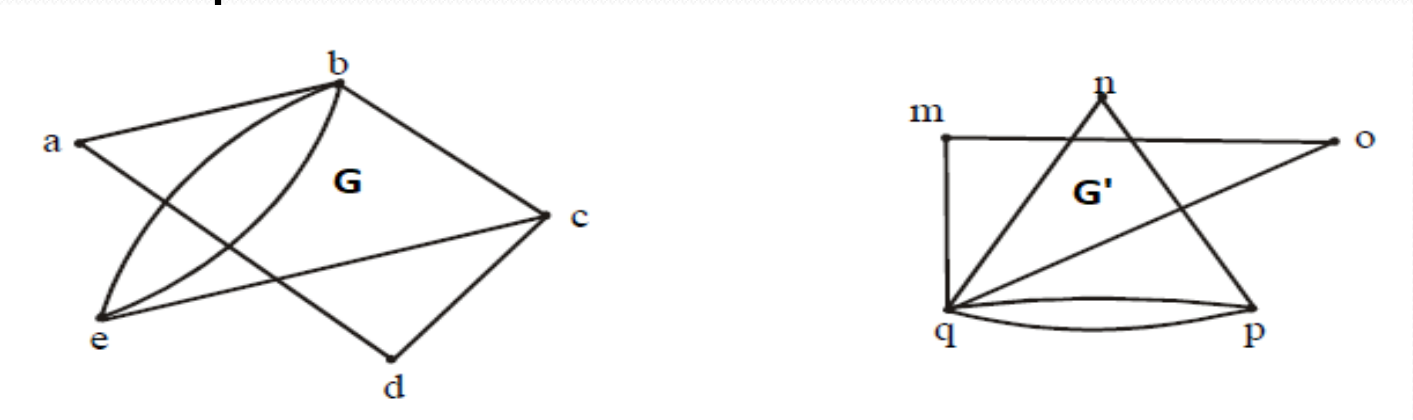
SOLUTION:

As both the graphs have the same number of vertices. But the graph G has 7 edges and the graph G' has only 6 edges. Therefore the two graphs are not isomorphic.

Note: As the edges of both the graphs G and G' are not same then how the one-one correspondence is possible ,that the reason the graphs G and G' are not isomorphic.

Isomorphism of Graphs

EXAMPLE: Determine whether the graph G and G' given below are isomorphic.

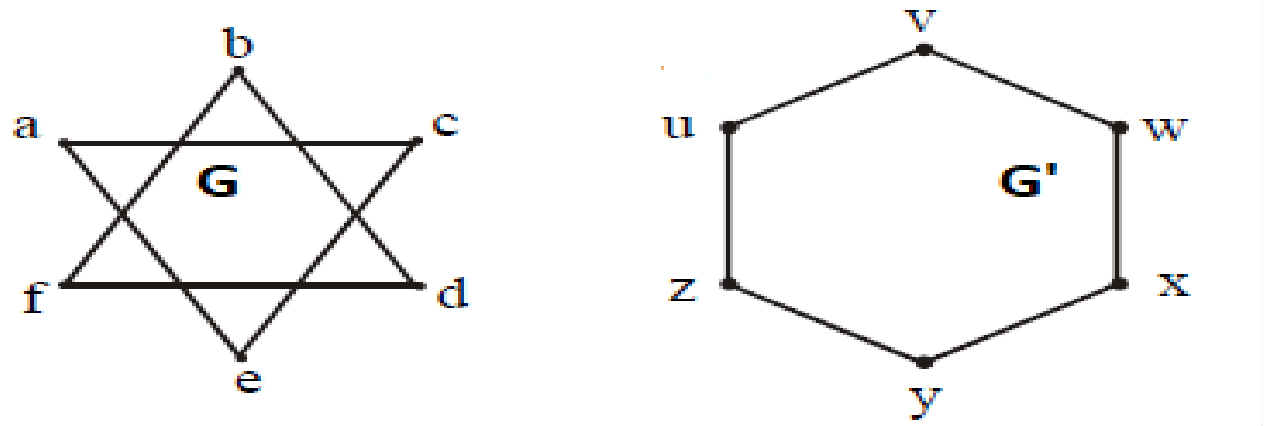


SOLUTION:

Both the graphs have 5 vertices and 7 edges. The vertex q of G' has degree 5. However G does not have any vertex of degree 5 (so one-one correspondence is not possible). Hence, the two graphs are not isomorphic.

Isomorphism of Graphs

EXAMPLE: Determine whether the graph G and G' given below are isomorphic.

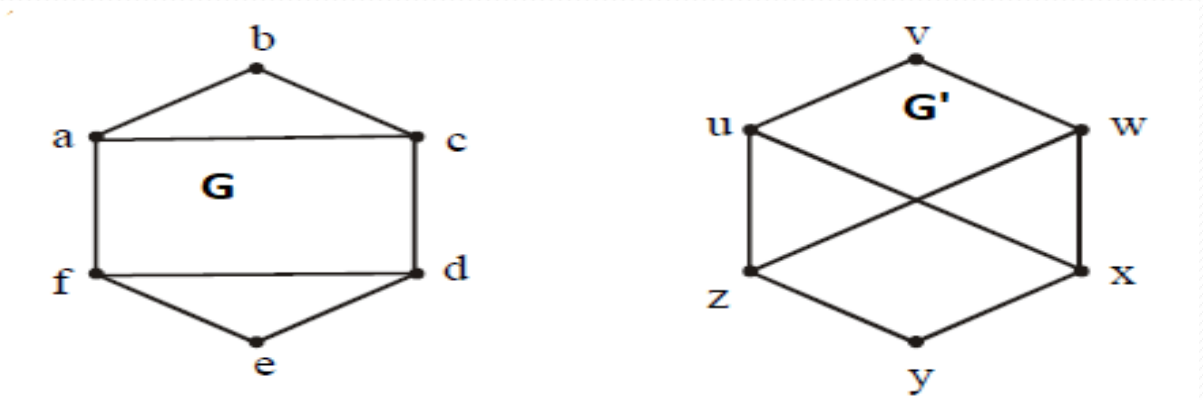


SOLUTION:

Clearly the vertices of both the graphs G and G' have the same degree (i.e "2") and having the same number of vertices and edges but isomorphism is not possible. As the graph G' is a connected graph but the graph G is not connected due to have two components (eca and bdf). Therefore the two graphs are non isomorphic.

Isomorphism of Graphs

EXAMPLE: Determine whether the graph G and G' given below are isomorphic.



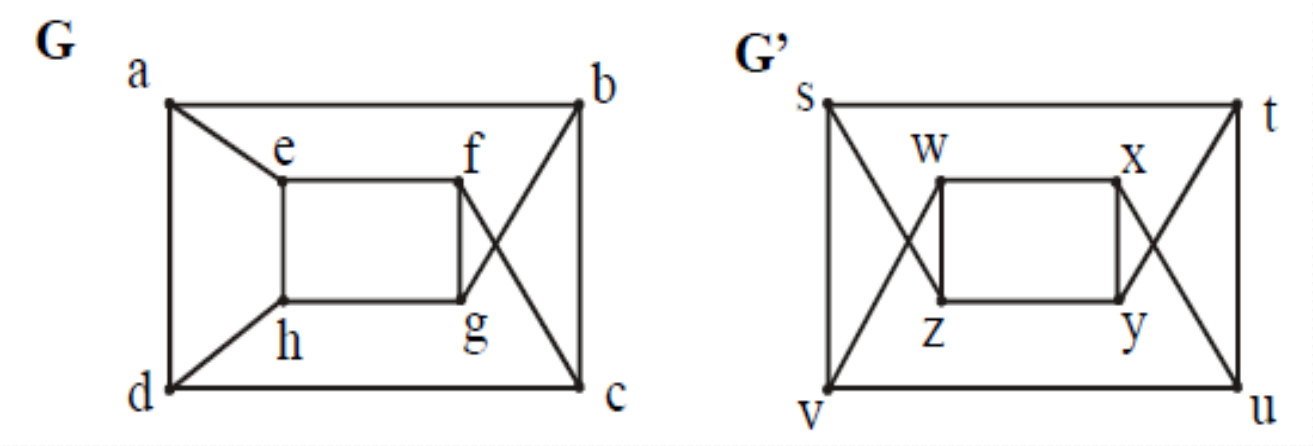
SOLUTION:

Clearly G has six vertices, G' also has six vertices. And the graph G has two simple circuits of length 3; one is abca and the other is defd. But G' does not have any simple circuit of length 3(as one simple circuit in G' is uxwv of length 4).Therefore the two graphs are non-isomorphic.

Note: A simple circuit is a circuit that does not have any other repeated vertex except the first and last.

Isomorphism of Graphs

EXAMPLE: Determine whether the graph G and G' given below are isomorphic.



SOLUTION:

Both the graph G and G' have 8 vertices and 12 edges and both are also called regular graph(as each vertex has degree 3).The graph G has two simple circuits of length 5; abcfea(i.e starts and ends at a) and cdhgfc(i.e starts and ends at c). But G' does not have any simple circuit of length 5 (it has simple circuit tyxut,vwxuv of length 4 etc). Therefore the two graphs are non-isomorphic.

Isomorphism of Graphs

EXAMPLE: Determine whether the given graph G and H are isomorphic.

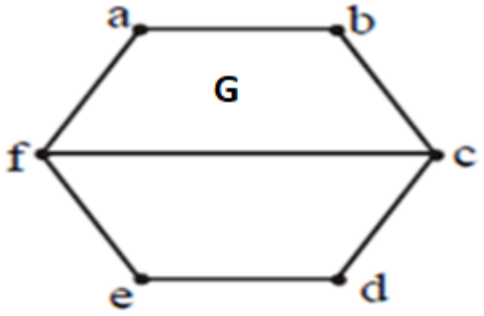
Solution:

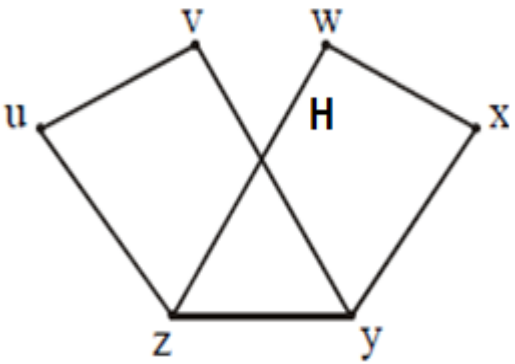
Solution: The function f with

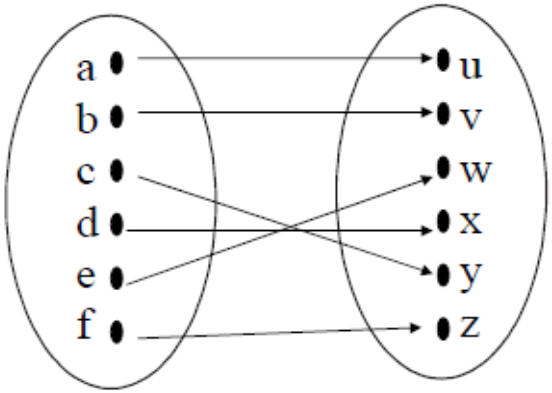
f(a) = u, f(b) = v, f(c) = y, f(d) = x, f(e) = w and f(f) = z is a one-to-one correspondence between G and H.

Note that adjacent vertices in G are a and b, b and c, c and d, c and f, d and e, e and f and f and a.

Each of the pairs f(a) = u and f(b) = v, f(b) = v and f(c) = y , f(c) = y and f(d) = x , f(c) = y and f(f) = z , f(d) = x and f(e) = w, f(e) = w and f(f) = z and f(f) = z and f(a) = u consists of two adjacent vertices in H.

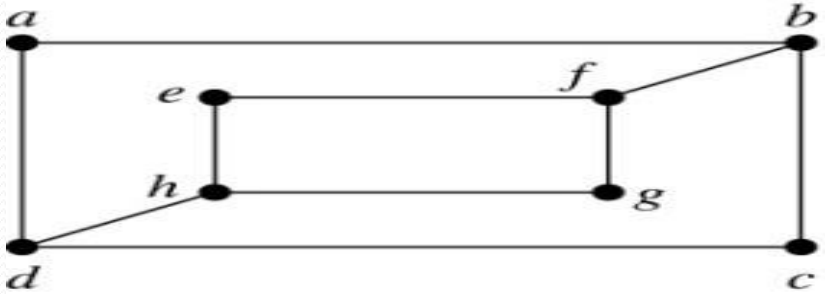




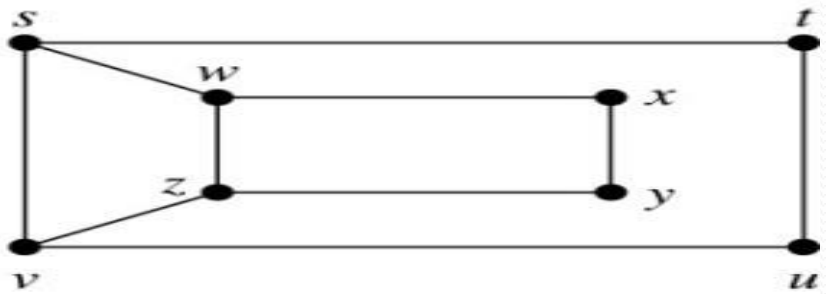


Isomorphism of Graphs (*cont.*)

Example: Determine whether these two graphs are isomorphic.



G

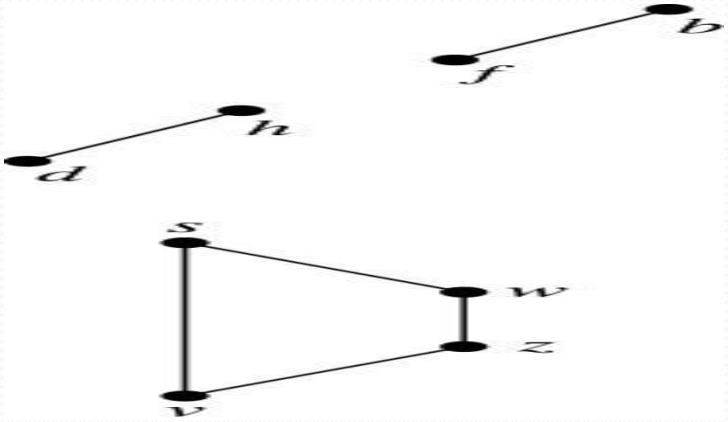


H

Solution: Both graphs have eight vertices and ten edges.
They also both have four vertices of degree two and four of degree three.

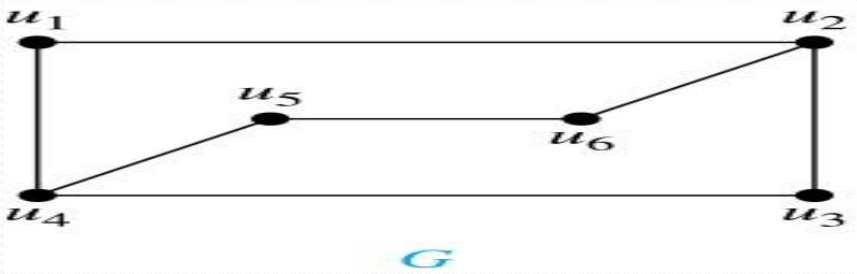
However, *G* and *H* are not isomorphic. Note that since $deg(a) = 2$ in *G*, *a* must correspond to *t*, *u*, *x*, or *y* in H, because these are the vertices of degree 2. But each of these vertices is adjacent to another vertex of degree two in *H*, which is not true for *a* in *G*.

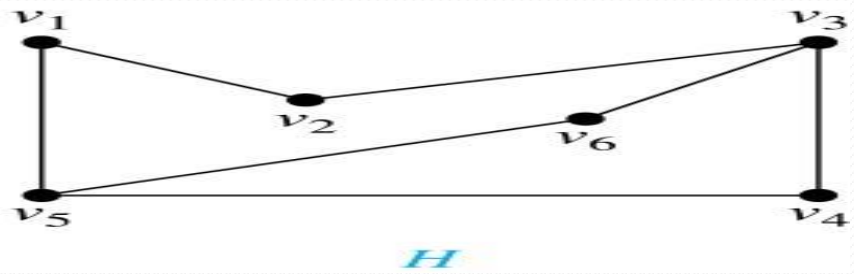
Alternatively, note that the subgraphs of *G* and *H* made up of vertices of degree three and the edges connecting them must be isomorphic.
But the subgraphs, as shown at the right, are not isomorphic.



Isomorphism of Graphs

Example: Determine whether these two graphs are isomorphic.





Solution: Both graphs have six vertices and seven edges.

They also both have four vertices of degree two and two of degree three. The subgraphs of G and H consisting of all the vertices of degree two and the edges connecting them are isomorphic. So, it is reasonable to try to find an isomorphism f.

We define an injection f from the vertices of G to the vertices of H that preserves the degree of vertices. We will determine whether it is an isomorphism.

The function f with $f(u_1) = v_6$, $f(u_2) = v_3$, $f(u_3) = v_4$, and $f(u_4) = v_5$, $f(u_5) = v_1$, and $f(u_6) = v_2$ is a one-to-one correspondence between G and H. Showing that this correspondence preserves edges is straightforward, so we will omit the details here. Because f is an isomorphism, it follows that G and H are isomorphic graphs.

Algorithms for Graph Isomorphism

- The best algorithms known for determining whether two graphs are isomorphic have exponential worst-case time complexity (in the number of vertices of the graphs).
- However, there are algorithms with linear average-case time complexity.
- You can use a public domain program called NAUTY to determine in less than a second whether two graphs with as many as 100 vertices are isomoprhic.
- Graph isomorphism is a problem of special interest because it is one of a few NP problems not known to be either tractable or NP-complete (see Section 3.3).

Applications of Graph Isomorphism

- The question whether graphs are isomorphic plays an important role in applications of graph theory. For example,
  - chemists use molecular graphs to model chemical compounds. Vertices represent atoms and edges represent chemical bonds. When a new compound is synthesized, a database of molecular graphs is checked to determine whether the graph representing the new compound is isomorphic to the graph of a compound that this already known.
  - Electronic circuits are modeled as graphs in which the vertices represent components and the edges represent connections between them. Graph isomorphism is the basis for
    - the verification that a particular layout of a circuit corresponds to the design's original schematics.
    - determining whether a chip from one vendor includes the intellectual property of another vendor.

Connectivity

Section 10.4

Section Summary

- Paths
- Connectedness in Undirected Graphs
- Connectedness in Directed Graphs
- Counting Paths between Vertices

Paths

Informal Definition: A *path* is a sequence of edges that begins at a vertex of a graph and travels from vertex to vertex along edges of the graph. As the path travels along its edges, it visits the vertices along this path, that is, the endpoints of these.

Applications: Numerous problems can be modeled with paths formed by traveling along edges of graphs such as:

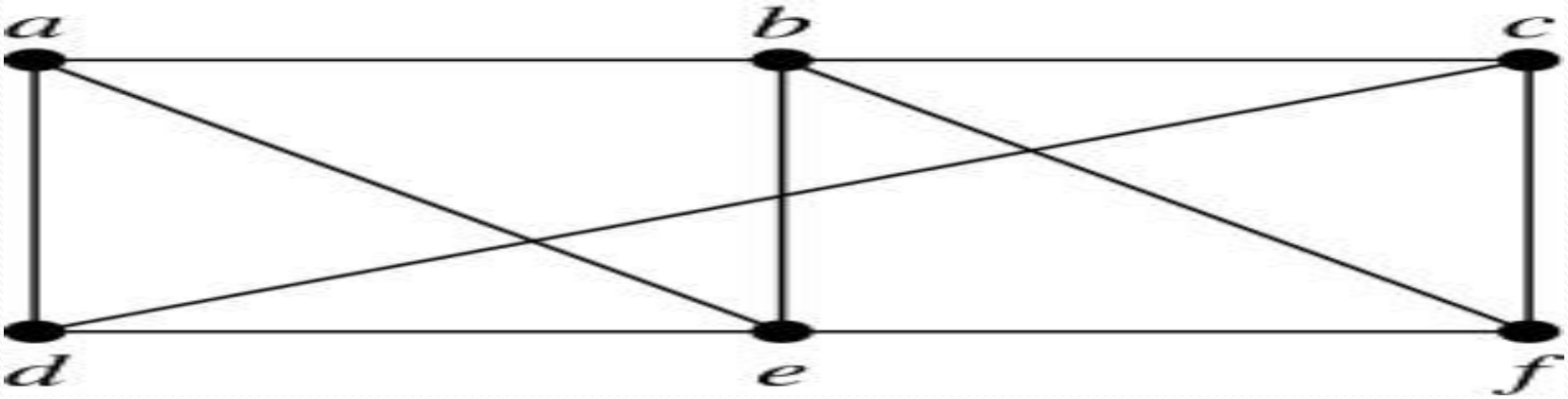
- determining whether a message can be sent between two computers.
- efficiently planning routes for mail delivery.

Paths

Definition: Let n be a nonnegative integer and G an undirected graph. A *path* of *length* n from u to v in G is a sequence of n edges $e_1, \ldots, e_n$ of G for which there exists a sequence $x_0 = u, x_1, \ldots, x_{n-1}, x_n = v$ of vertices such that e_i has, for $i = 1, \ldots, n$, the endpoints x_{i-1} and x_i.

- When the graph is simple, we denote this path by its vertex sequence $x_0, x_1, \ldots, x_n$ (since listing the vertices uniquely determines the path).
- The path is a *circuit* if it begins and ends at the same vertex ($u = v$) and has length greater than zero.
- The path or circuit is said to *pass through* the vertices $x_1, x_2, \ldots, x_{n-1}$ and *traverse* the edges $e_1, \ldots, e_n$.
- A path or circuit is *simple* if it does not contain the same edge more than once.

Paths (*continued*)



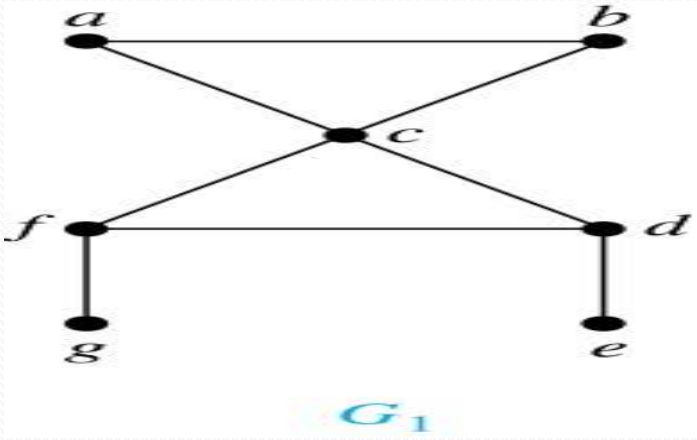
Example: In the simple graph here:

- *a*, *d*, *c*, *f*, *e* is a simple path of length 4.
- *d*, *e*, *c*, *a* is not a path because *e* is not connected to *c*.
- *b*, *c*, *f*, *e*, *b* is a circuit of length 4.
- *a*, *b*, *e*, *d*, *a*, *b* is a path of length 5, but it is not a simple path.

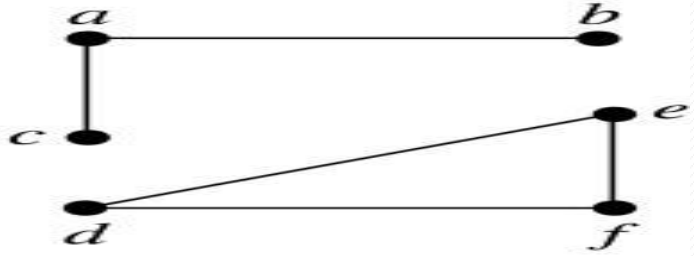
Connectedness in Undirected Graphs

Definition: An undirected graph is called *connected* if there is a path between every pair of vertices. An undirected graph that is not *connected* is called *disconnected*. We say that we *disconnect* a graph when we remove vertices or edges, or both, to produce a disconnected subgraph.

Example: G_1 is connected because there is a path between any pair of its vertices, as can be easily seen. However G_2 is not connected because there is no path between vertices *a* and *f*, for example.



G_1



G_2

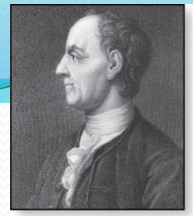
Euler and Hamiltonian Graphs

Section 10.5

Section Summary

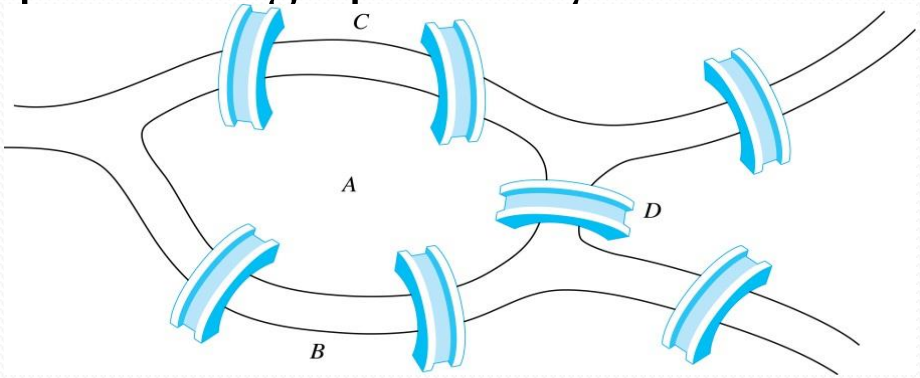
- Euler Paths and Circuits
- Hamilton Paths and Circuits
- Applications of Hamilton Circuits

Euler Paths and Circuits

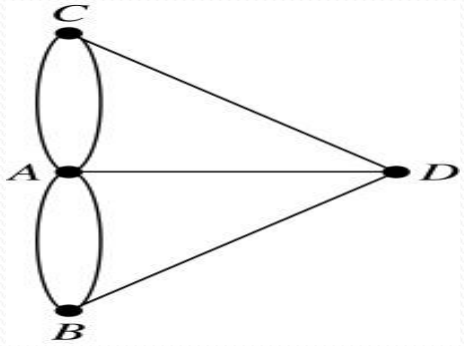


Leonard Euler
(1707-1783)

- The town of Königsberg, Prussia (now Kalingrad, Russia) was divided into four sections by the branches of the Pregel river. In the 18th century seven bridges connected these regions.
- People wondered whether whether it was possible to follow a path that crosses each bridge exactly once and returns to the starting point.
- The Swiss mathematician Leonard Euler proved that no such path exists. This result is often considered to be the first theorem ever proved in graph theory.



The 7 Bridges of Königsberg

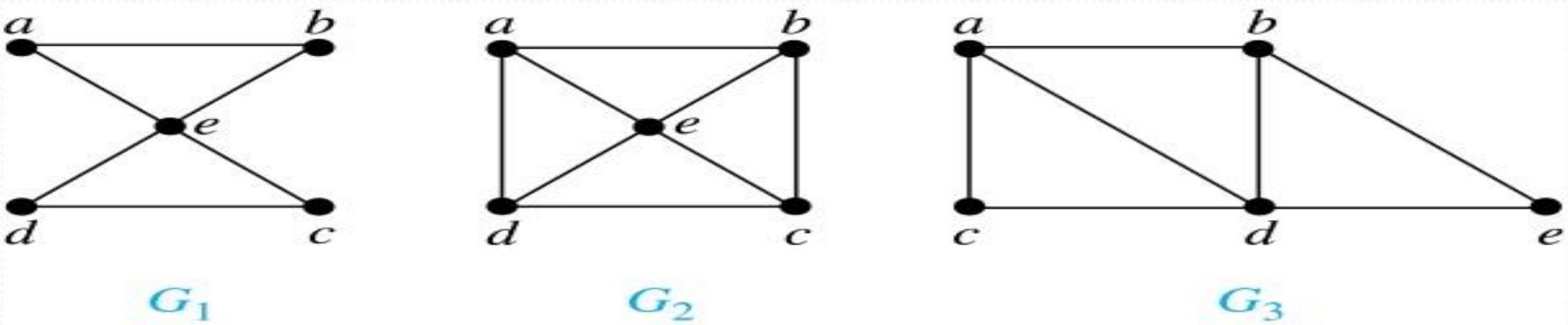


Multigraph Model of the Bridges of Königsberg

Euler Paths and Circuits

Definition: An *Euler circuit* in a graph G is a simple circuit containing every edge of G. An *Euler path* in G is a simple path containing every edge of G.

Example: Which of the undirected graphs G_1, G_2, and G_3 has a Euler circuit? Of those that do not, which has an Euler path?



Solution: The graph G_1 has an Euler circuit (e.g., a, e, c, d, e, b, a). But, as can easily be verified by inspection, neither G_2 nor G_3 has an Euler circuit. Note that G_3 has an Euler path (e.g., a, c, d, e, b, d, a, b), but there is no Euler path in G_2, which can be verified by inspection.

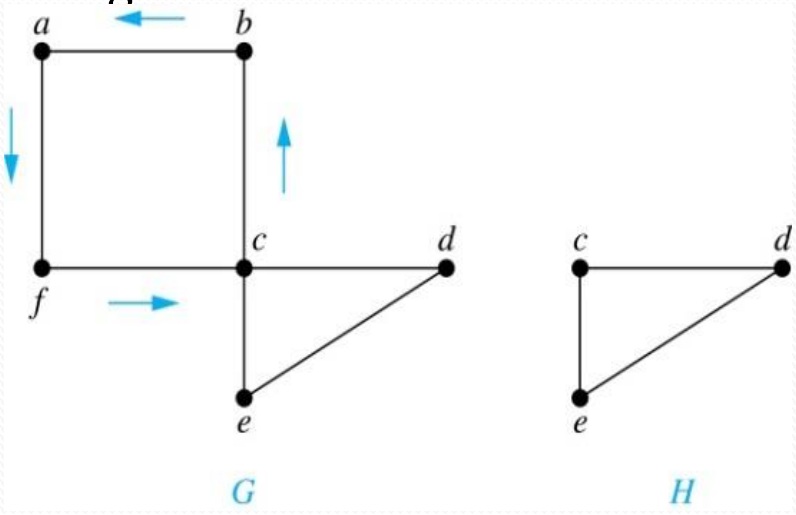
Necessary Conditions for Euler Circuits and Paths

- An Euler circuit begins with a vertex a and continues with an edge incident with a, say $\{a, b\}$. The edge $\{a, b\}$ contributes one to $\deg(a)$.
- Each time the circuit passes through a vertex it contributes two to the vertex's degree.
- Finally, the circuit terminates where it started, contributing one to $\deg(a)$. Therefore $\deg(a)$ must be even.
- We conclude that the degree of every other vertex must also be even.
- By the same reasoning, we see that the initial vertex and the final vertex of an Euler path have odd degree, while every other vertex has even degree. So, a graph with an Euler path has exactly two vertices of odd degree.
- In the next slide we will show that these necessary conditions are also sufficient conditions.

Sufficient Conditions for Euler Circuits and Paths

Suppose that G is a connected multigraph with ≥ 2 vertices, all of even degree. Let $x_0 = a$ be a vertex of even degree. Choose an edge $\{x_0, x_1\}$ incident with a and proceed to build a simple path $\{x_0, x_1\}, \{x_1, x_2\}, \ldots, \{x_{n-1}, x_n\}$ by adding edges one by one until another edge can not be added.

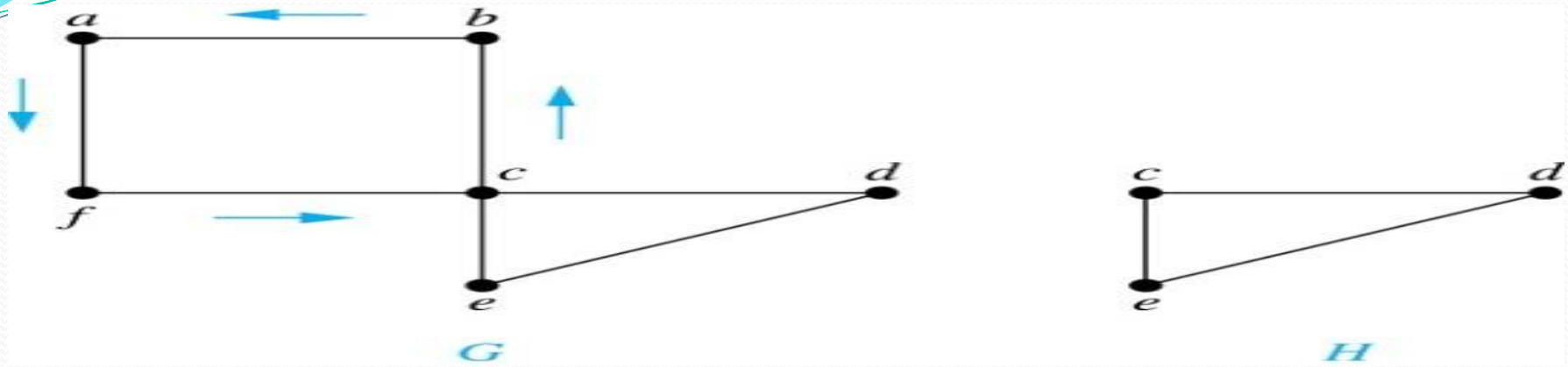
We illustrate this idea in the graph G here. We begin at a and choose the edges $\{a, f\}, \{f, c\}, \{c, b\}$, and $\{b, a\}$ in succession.



- The path begins at a with an edge of the form $\{a, x\}$; we show that it must terminate at a with an edge of the form $\{y, a\}$. Since each vertex has an even degree, there must be an even number of edges incident with this vertex. Hence, every time we enter a vertex other than a, we can leave it. Therefore, the path can only end at a.
- If all of the edges have been used, an Euler circuit has been constructed. Otherwise, consider the subgraph H obtained from G by deleting the edges already used.

In the example H consists of the vertices c, d, e.

Sufficient Conditions for Euler Circuits and Paths



- Because G is connected, H must have at least one vertex in common with the circuit that has been deleted.

In the example, the vertex is *c*.

- Every vertex in H must have even degree because all the vertices in *G* have even degree and for each vertex, pairs of edges incident with this vertex have been deleted. Beginning with the shared vertex construct a path ending in the same vertex (as was done before). Then splice this new circuit into the original circuit.

In the example, we end up with the circuit *a, f, c, d, e, c, b, a*.

- Continue this process until all edges have been used. This produces an Euler circuit. Since every edge is included and no edge is included more than once.
- Similar reasoning can be used to show that a graph with exactly two vertices of odd degree must have an Euler path connecting these two vertices of odd degree

Algorithm for Constructing an Euler Circuits

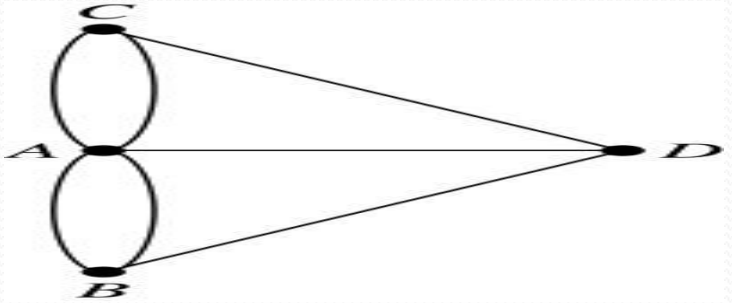
In our proof we developed this algorithms for constructing a Euler circuit in a graph with no vertices of odd degree.

procedure *Euler*(*G*: connected multigraph with all vertices of even degree)
circuit := a circuit in *G* beginning at an arbitrarily chosen vertex with edges successively added to form a path that returns to this vertex.
H := *G* with the edges of this circuit removed
while *H* has edges
subciruit := a circuit in *H* beginning at a vertex in *H* that also is an endpoint of an edge in circuit.
H := *H* with edges of *subciruit* and all isolated vertices removed
circuit := *circuit* with *subcircuit* inserted at the appropriate vertex.
return *circuit*{*circuit* is an Euler circuit}

Necessary and Sufficient Conditions for Euler Circuits and Paths (*continued*)

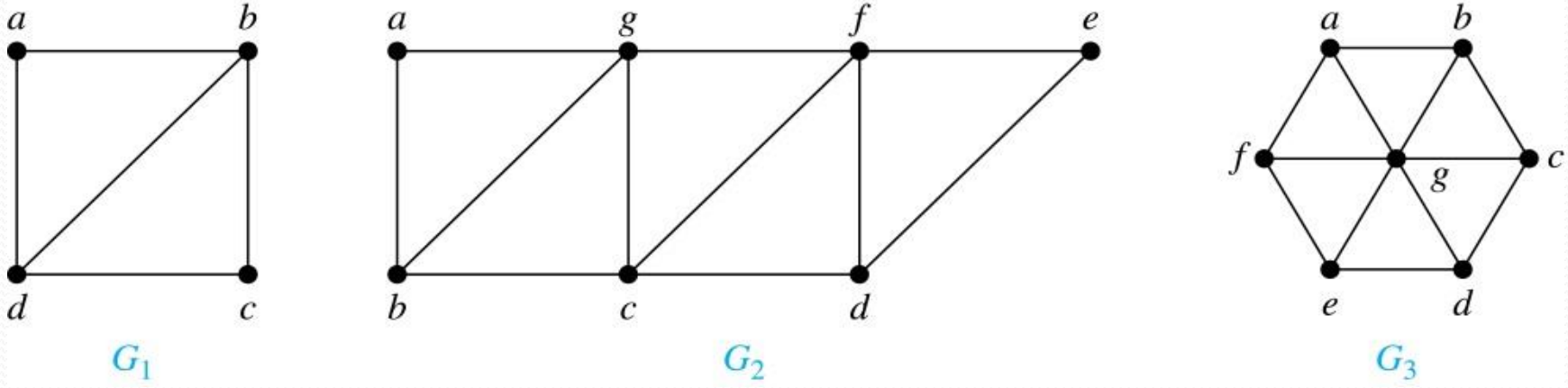
Theorem: A connected multigraph with at least two vertices has an Euler circuit if and only if each of its vertices has an even degree and it has an Euler path if and only if it has exactly two vertices of odd degree.

Example: Two of the vertices in the multigraph model of the Königsberg bridge problem have odd degree. Hence, there is no Euler circuit in this multigraph and it is impossible to start at a given point, cross each bridge exactly once, and return to the starting point.



Euler Circuits and Paths

Example:



G_1 contains exactly two vertices of odd degree (b and d). Hence it has an Euler path, e.g., d, a, b, c, d, b.

G_2 has exactly two vertices of odd degree (b and d). Hence it has an Euler path, e.g., $b, a, g, f, e, d, c, g, b, c, f, d$.

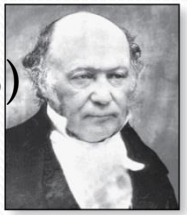
G_3 has six vertices of odd degree. Hence, it does not have an Euler path.

Applications of Euler Paths and Circuits

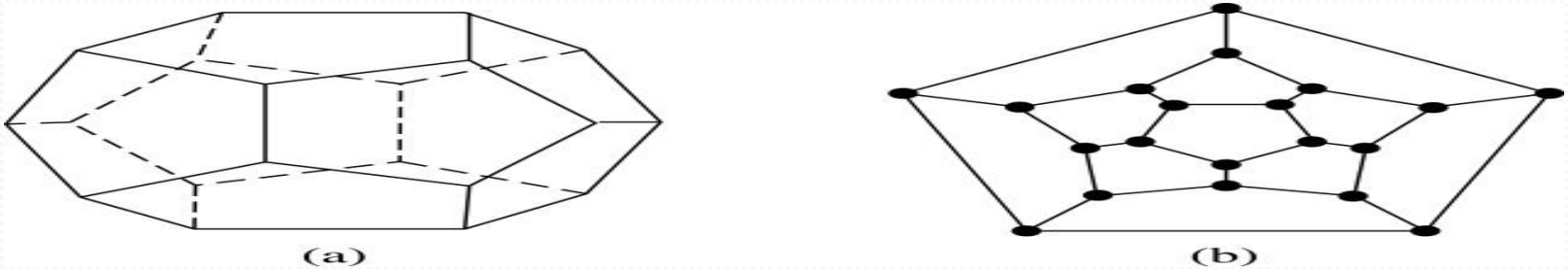
- Euler paths and circuits can be used to solve many practical problems such as finding a path or circuit that traverses each
  - street in a neighborhood,
  - road in a transportation network,
  - connection in a utility grid,
  - link in a communications network.
- Other applications are found in the
  - layout of circuits,
  - network multicasting,
  - molecular biology, where Euler paths are used in the sequencing of DNA.

Hamilton Paths and Circuits

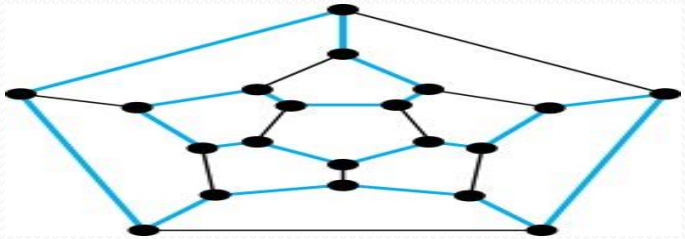
William Rowan Hamilton (1805- 1865)



- Euler paths and circuits contained every edge only once.
  Now we look at paths and circuits that contain every vertex exactly once.
- William Hamilton invented the *Icosian puzzle* in 1857. It consisted of a wooden dodecahedron (with 12 regular pentagons as faces), illustrated in (a), with a peg at each vertex, labeled with the names of different cities. String was used to used to plot a circuit visiting 20 cities exactly once
- The graph form of the puzzle is given in (b).



- The solution (a Hamilton circuit) is given here.



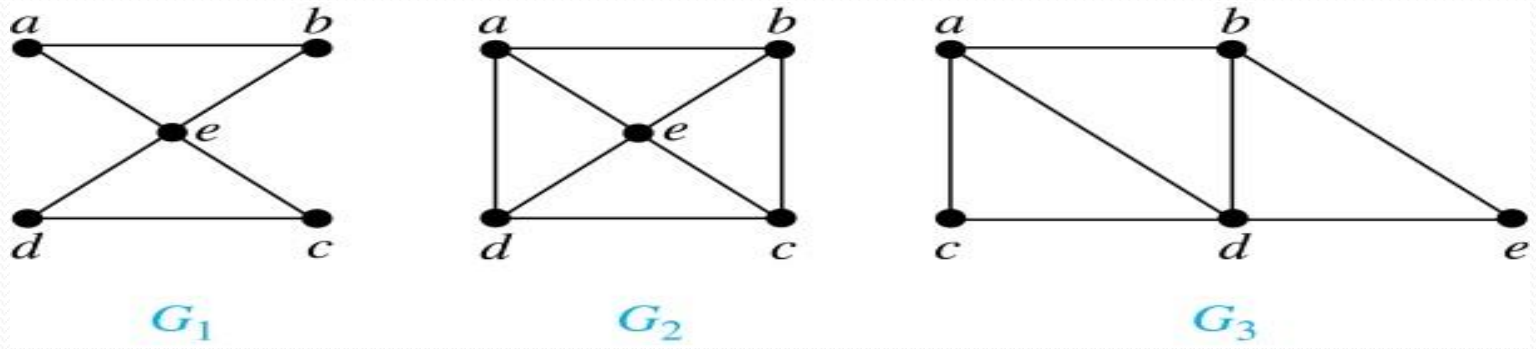
Hamilton Paths and Circuits

Definition: A simple path in a graph G that passes through every vertex exactly once is called a *Hamilton path*, and a simple circuit in a graph G that passes through every vertex exactly once is called a *Hamilton circuit*.

That is, a simple path $x_0, x_1, \ldots, x_{n-1}, x_n$ in the graph $G = (V, E)$ is called a Hamilton path if $V = \{x_0, x_1, \ldots, x_{n-1}, x_n\}$ and $x_i \neq x_j$ for $0 \leq i < j \leq n$, and the simple circuit $x_0, x_1, \ldots, x_{n-1}, x_n, x_0$ (with $n > 0$) is a Hamilton circuit if $x_0, x_1, \ldots, x_{n-1}, x_n$ is a Hamilton path.

Hamilton Paths and Circuits

Example: Which of these simple graphs has a Hamilton circuit or, if not, a Hamilton path?



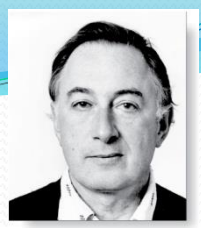
Solution:

G_1 does not have a Hamilton circuit (Why?), but does have a Hamilton path : a, b ,e , c, d.

G_2 has a Hamilton circuit: a, b, c, d, e, a.

G_3 has a Hamilton circuit: a,b,e,d,c,a

Necessary Conditions for Hamilton Circuits



Gabriel Andrew Dirac
(1925-1984)

- Unlike for an Euler circuit, no simple necessary and sufficient conditions are known for the existence of a Hamiton circuit.
- However, there are some useful necessary conditions. We describe two of these now.

  Dirac's Theorem: If G is a simple graph with $n \geq 3$ vertices such that the degree of every vertex in G is $\geq n/2$, then G has a Hamilton circuit.

  Ore's Theorem: If G is a simple graph with $n \geq 3$ vertices such that $\deg(u) + \deg(v) \geq n$ for every pair of nonadjacent vertices, then G has a Hamilton circuit.

Øysten Ore
(1899-1968)

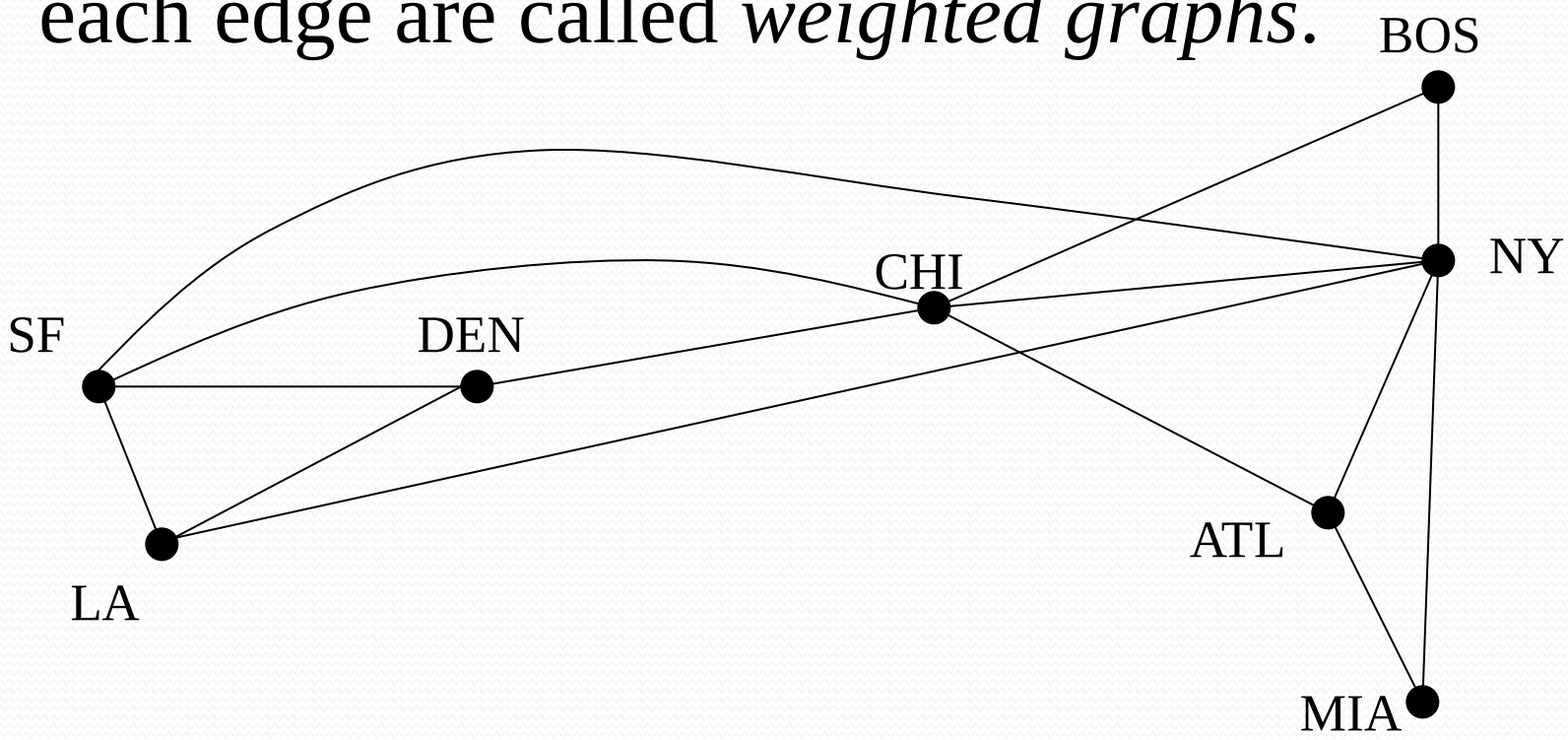
Applications of Hamilton Paths and Circuits

- Applications that ask for a path or a circuit that visits each intersection of a city, each place pipelines intersect in a utility grid, or each node in a communications network exactly once, can be solved by finding a Hamilton path in the appropriate graph.
- The famous *traveling salesperson problem* (*TSP*) asks for the shortest route a traveling salesperson should take to visit a set of cities. This problem reduces to finding a Hamilton circuit such that the total sum of the weights of its edges is as small as possible.
- A family of binary codes, known as *Gray codes*, which minimize the effect of transmission errors, correspond to Hamilton circuits in the n-cube Q_n.

Shortest Paths

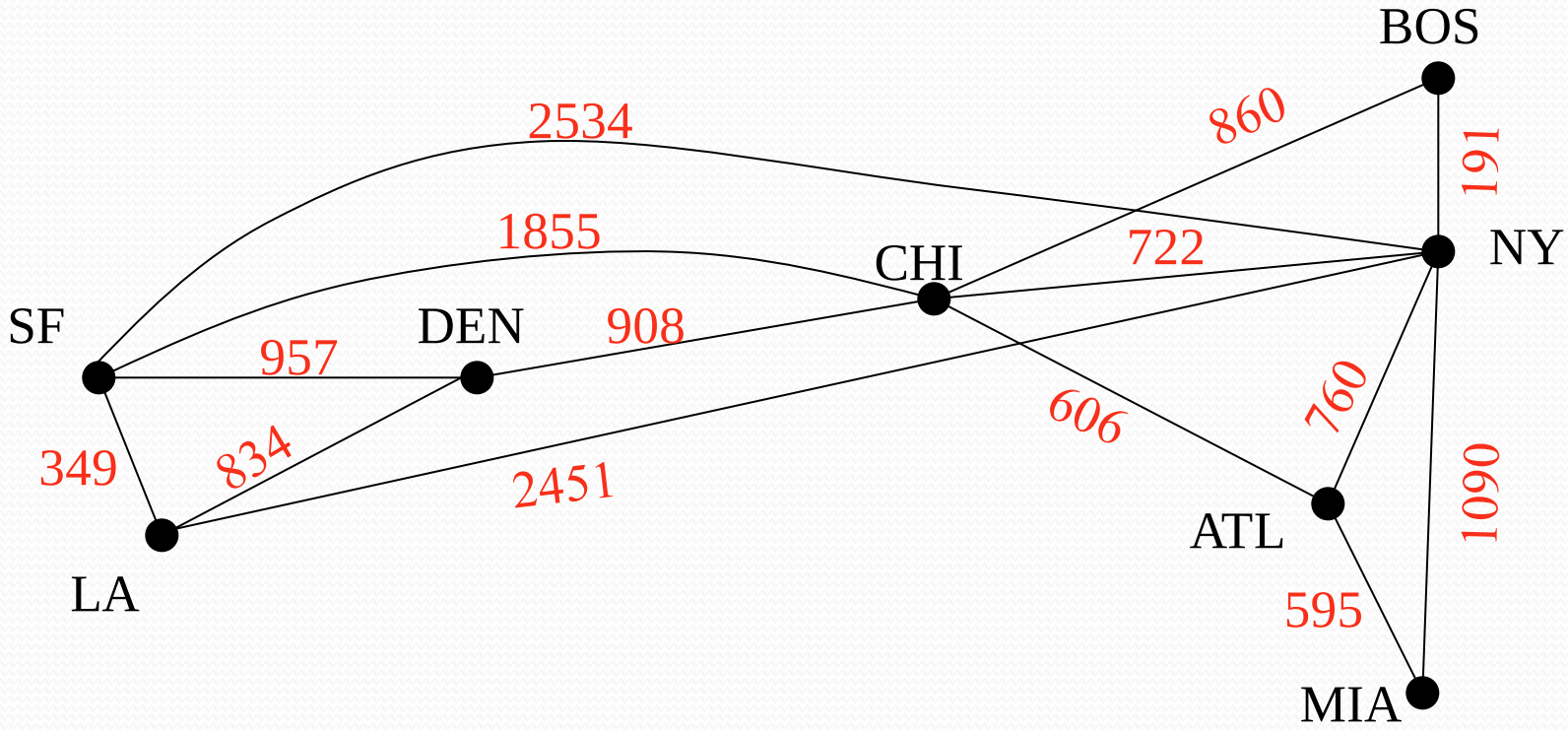
Weighted Graphs

Graphs that have a number assigned to each edge are called *weighted graphs*.



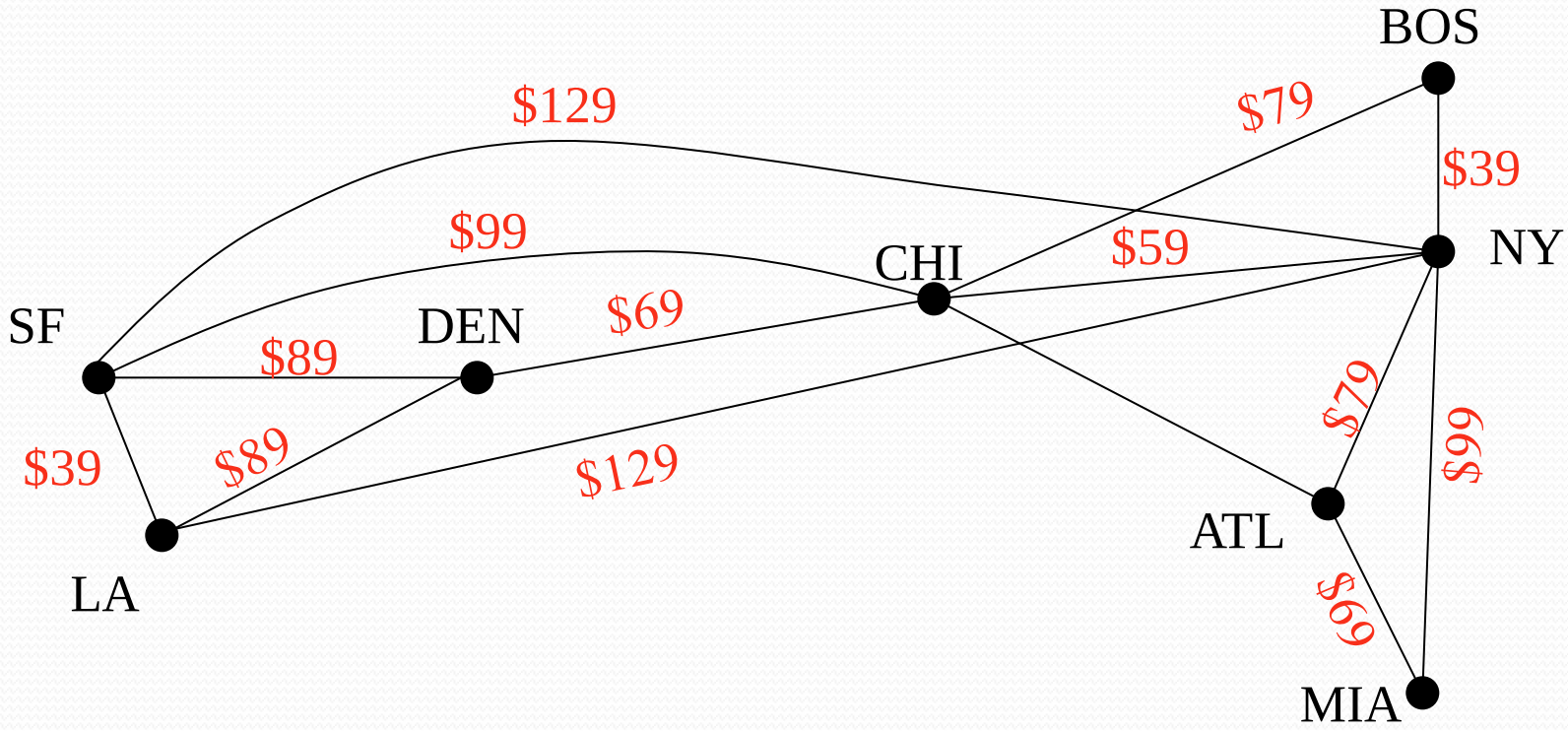
Weighted Graphs

MILES



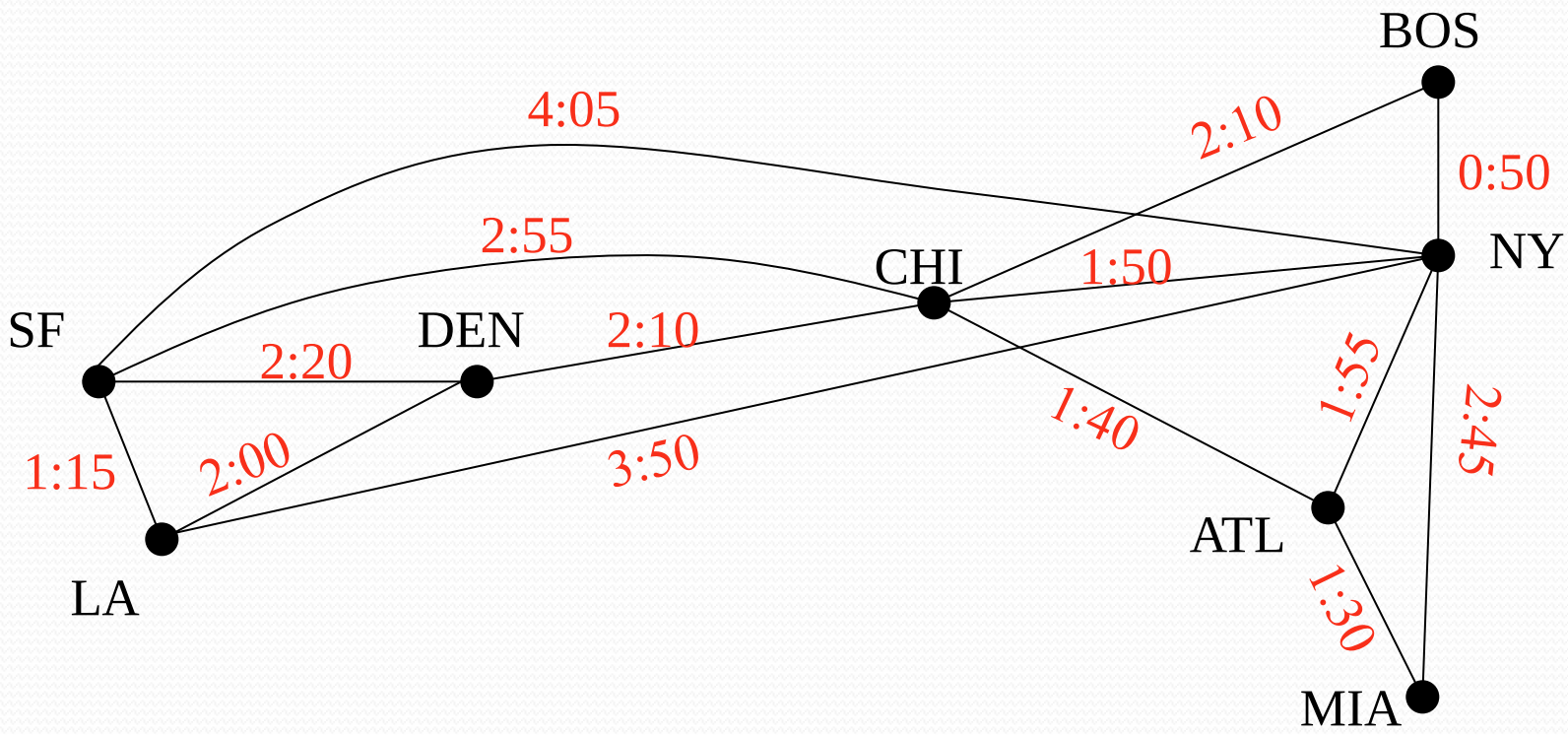
Weighted Graphs

FARES



Weighted Graphs

FLIGHT TIMES



Weighted Graphs

- A weighted graph is a graph in which each edge (u, v) has a weight w(u, v). Each weight is a real number.
- Weights can represent distance, cost, time, capacity, etc.
- The length of a path in a weighted graph is the sum of the weights on the edges.
- Dijkstra's Algorithm finds the shortest path between two vertices.

Dijkstra's Algorithm

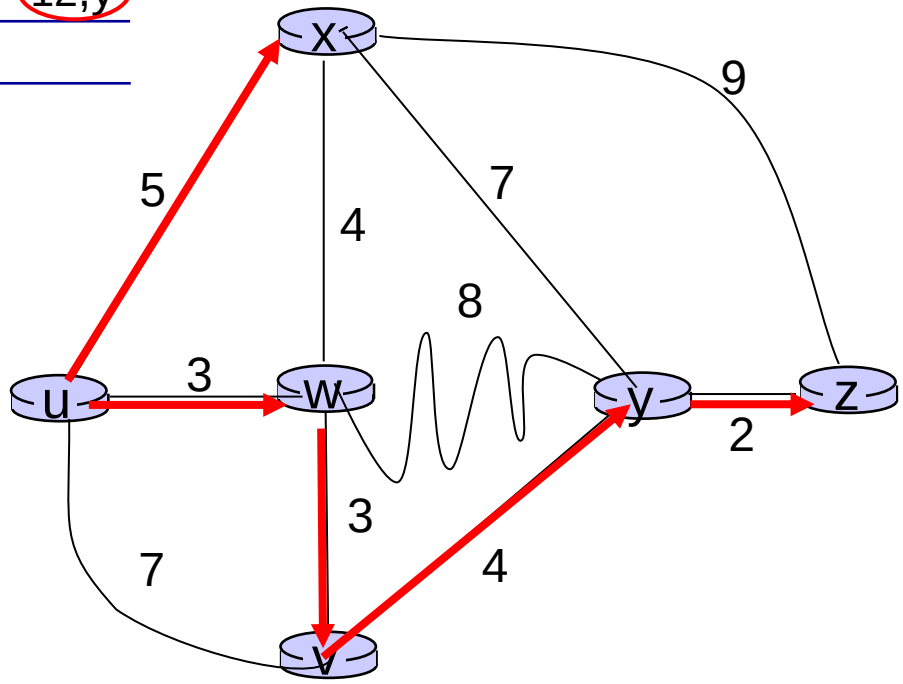
- Dijkstra's algorithm is used in problems relating to finding the shortest path.
- Each node is given a temporary label denoting the length of the shortest path *from* the start node *so far*.
- This label is replaced if another shorter route is found.
- Once it is certain that no other shorter paths can be found, the temporary label becomes a permanent label.
- Eventually all the nodes have permanent labels.
- At this point the shortest path is found by retracing the path backwards.

Dijkstra's algorithm: example

Step	N'	D(v) p(v)	D(w) p(w)	D(x) p(x)	D(y) p(y)	D(z) p(z)
0	u	7,u	3,u	5,u	∞	∞
1	uw	6,w		5,u	11,w	∞
2	uwx	6,w			11,w	14,x
3	uwxv				10,v	14,x
4	uwxvy					12,y
5	uwxvyz					

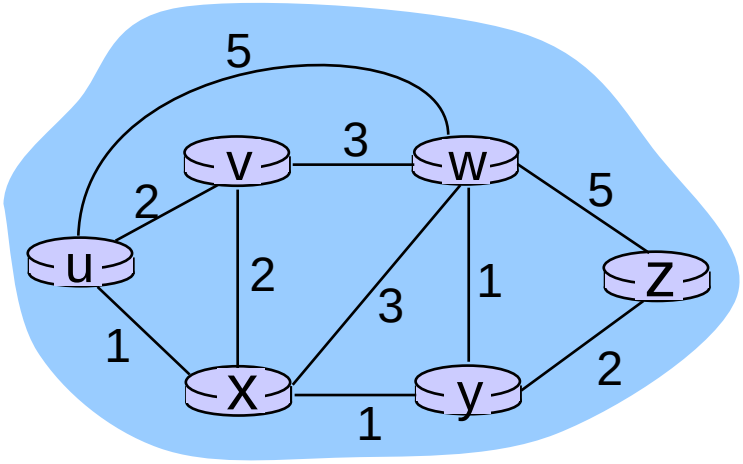
notes:

- construct shortest path tree by tracing predecessor nodes
- ties can exist (can be broken arbitrarily)



Dijkstra's algorithm: another example

Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



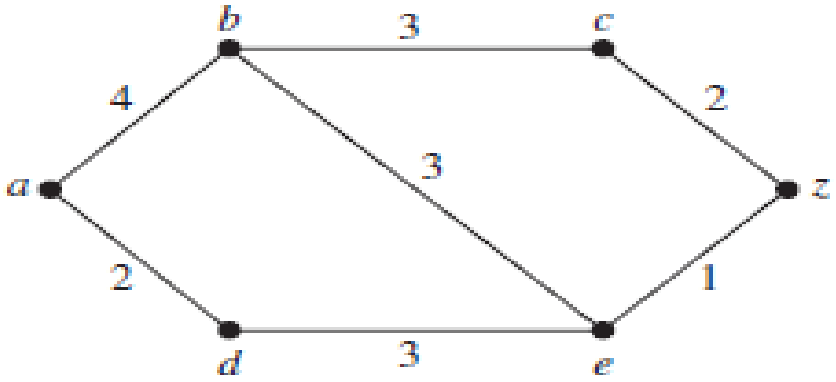
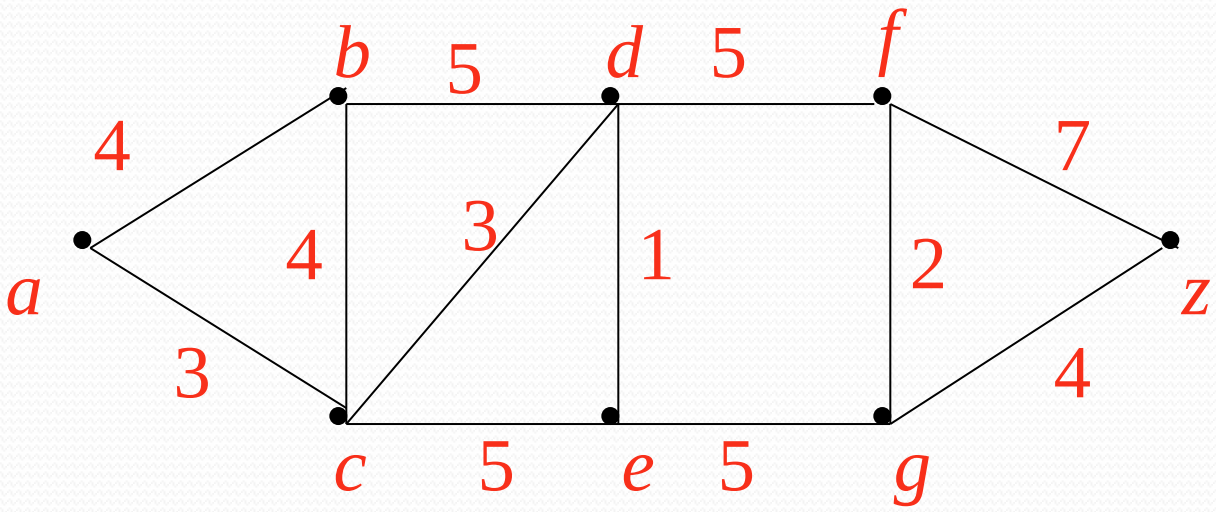


FIGURE 3 **A Weighted Simple Graph.**

What is the length of a shortest path between a and z in the weighted graph shown in Figure 3?

Problem: shortest path from a to z



The Traveling Salesman Problem

- The **traveling salesman problem** is one of the classical problems in computer science.
- A traveling salesman wants to visit a number of cities and then return to his starting point. Of course he wants to save time and energy, so he wants to determine the **shortest cycle** for his trip.
- We can represent the cities and the distances between them by a weighted, complete, undirected graph.
- The problem then is to find the **shortest cycle (of minimum total weight that visits each vertex exactly one)**.
- Finding the shortest cycle is different than Dijkstra's shortest path.
  It is much harder too, no polynomial time algorithm exists!

The Traveling Salesman Problem

- Importance:
  - Variety of scheduling application can be solved as a traveling salesmen problem.
  - Examples:
    - Ordering drill position on a drill press.
    - School bus routing.
  - The problem has theoretical importance because it represents a class of difficult problems known as NP-hard problems.

Travelling Salesman problem

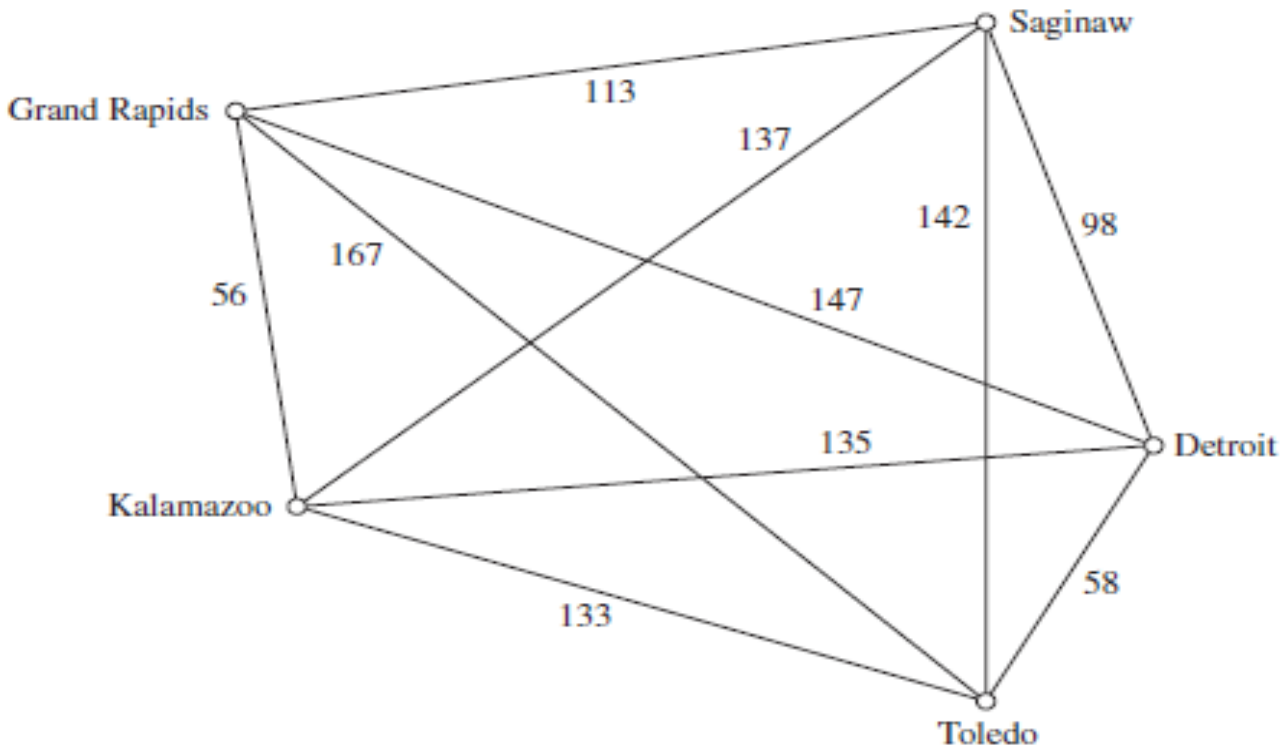


FIGURE 5 **The Graph Showing the Distances between Five Cities.**

Travelling Salesman problem

Route	*Total Distance (miles)*
Detroit–Toledo–Grand Rapids–Saginaw–Kalamazoo–Detroit	610
Detroit–Toledo–Grand Rapids–Kalamazoo–Saginaw–Detroit	516
Detroit–Toledo–Kalamazoo–Saginaw–Grand Rapids–Detroit	588
Detroit–Toledo–Kalamazoo–Grand Rapids–Saginaw–Detroit	458
Detroit–Toledo–Saginaw–Kalamazoo–Grand Rapids–Detroit	540
Detroit–Toledo–Saginaw–Grand Rapids–Kalamazoo–Detroit	504
Detroit–Saginaw–Toledo–Grand Rapids–Kalamazoo–Detroit	598
Detroit–Saginaw–Toledo–Kalamazoo–Grand Rapids–Detroit	576
Detroit–Saginaw–Kalamazoo–Toledo–Grand Rapids–Detroit	682
Detroit–Saginaw–Grand Rapids–Toledo–Kalamazoo–Detroit	646
Detroit–Grand Rapids–Saginaw–Toledo–Kalamazoo–Detroit	670
Detroit–Grand Rapids–Toledo–Saginaw–Kalamazoo–Detroit	728