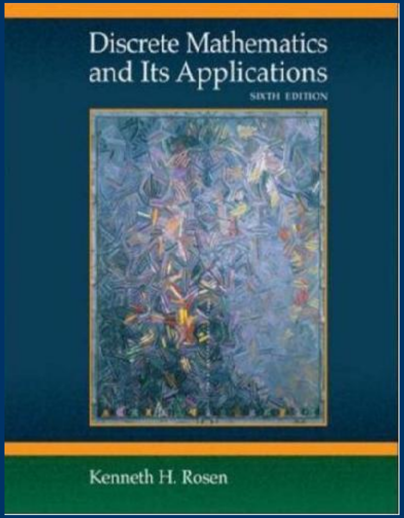
Chapter 9 (Part 2): Graphs

- Graph Terminology (9.2) (cont.)
- Representing Graphs & Graph Isomorphism (9.3)
- Connectivity (9.4)



Graph Terminology (9.2) (cont.)

- Some special simple graphs

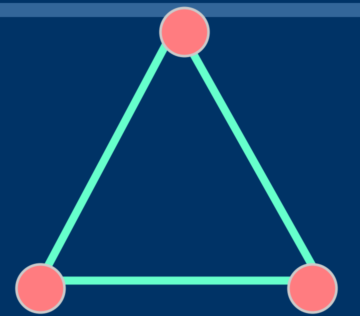
  – Complete graph

    They are denoted by K_n, they contain exactly one edge between each pair of distinct vertices

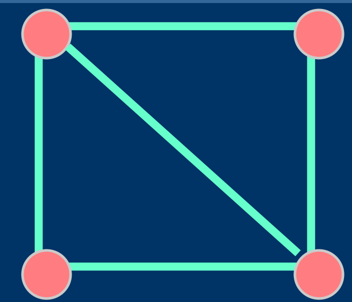
Graph Terminology (9.2) (cont.)

K_1

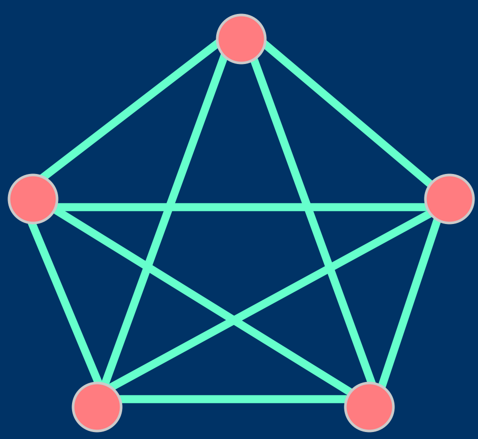
K_2



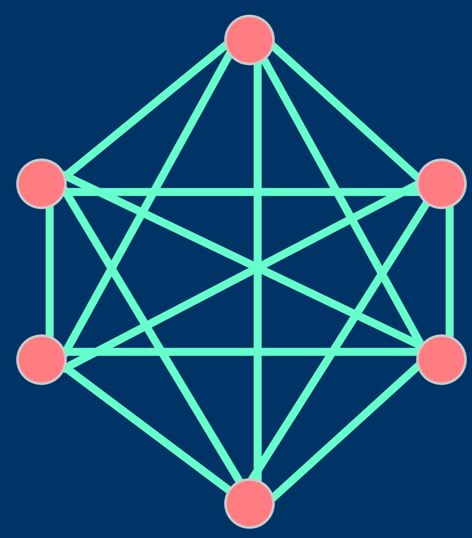
K_3



K_4



K_5



K_6

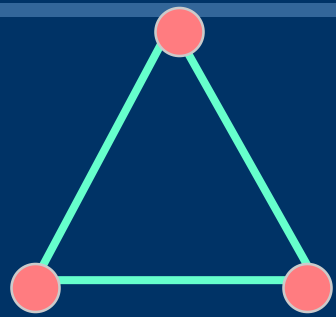
The graphs K_n for $1 \leq n \leq 6$

Graph Terminology (9.2) (cont.)

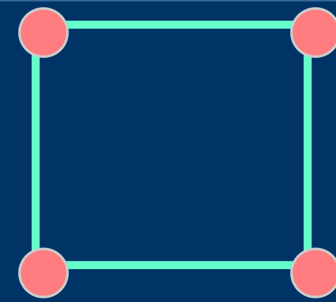
– Cycles

They are denoted by $C_n (n \geq 3)$: they consist of n vertices $v_1, v_2, \ldots, v_n$ and edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_n, v_{n-1}\}$ and $\{v_n, v_1\}$

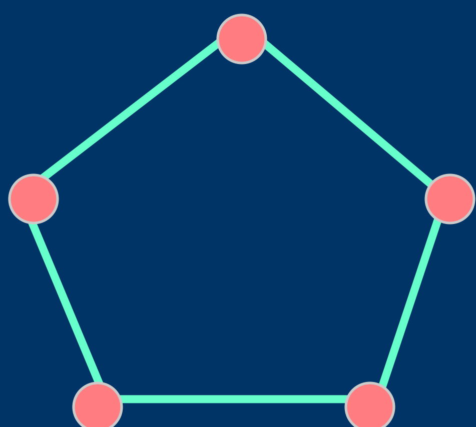
Graph Terminology (9.2) (cont.)



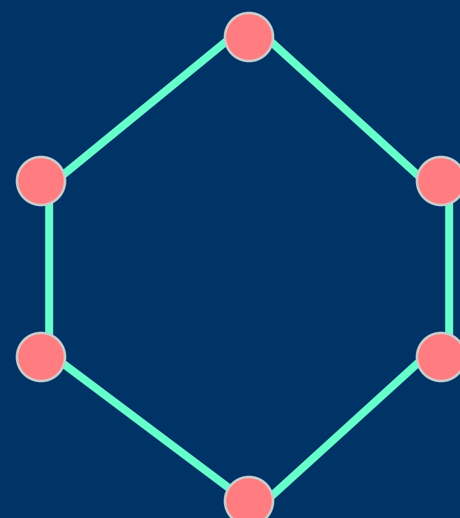
C_3



C_4



C_5



C_6

The cycles C_3, C_4, C_5 & C_6

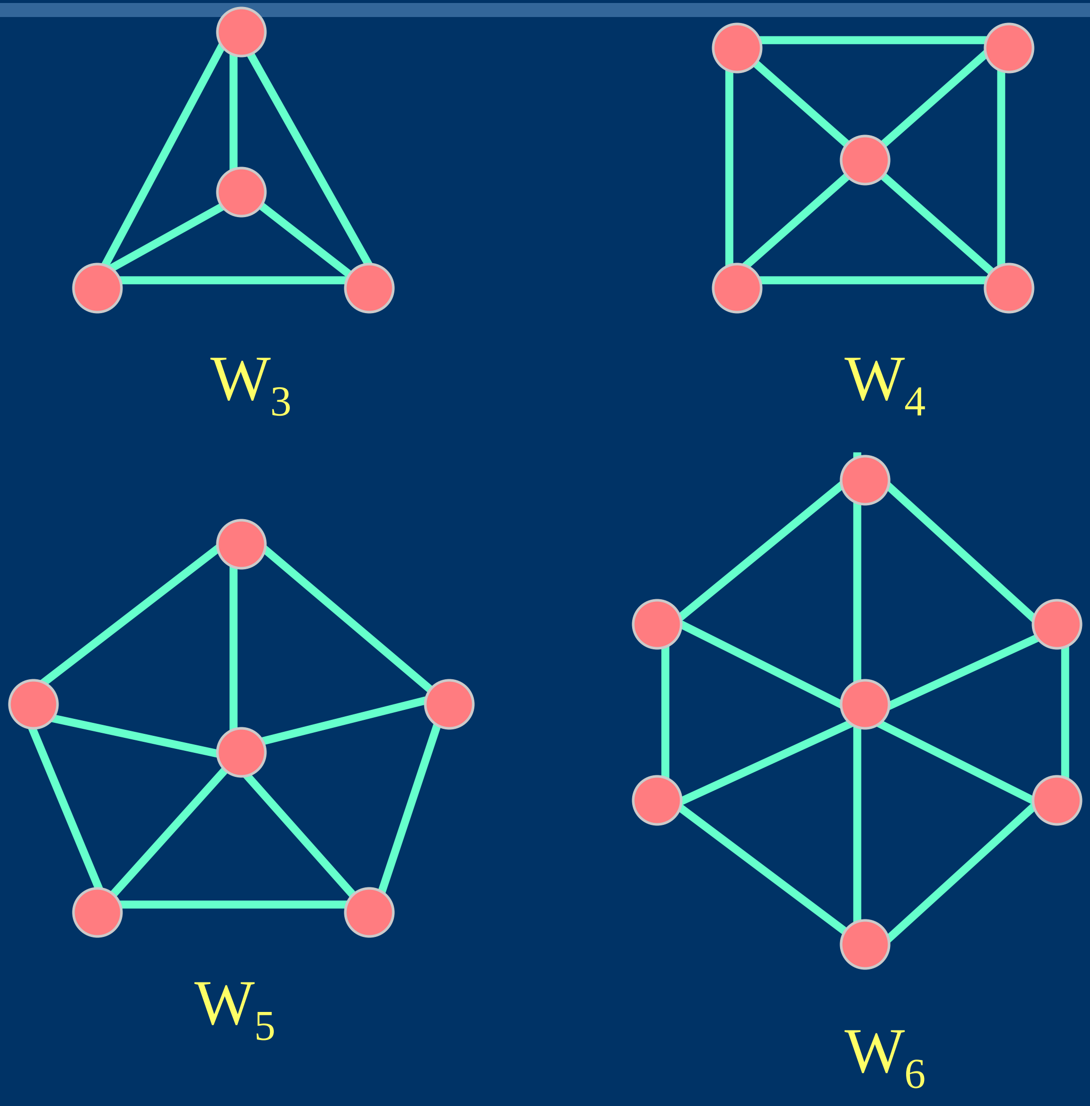
Graph Terminology (9.2) (cont.)

- Wheels

  They are denoted by W_n; they are obtained by adding a vertex to the graphs C_n and connect this vertex to all vertices

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

Graph Terminology (9.2) (cont.)



The Wheels W_3, W_4, W_5 & W_6

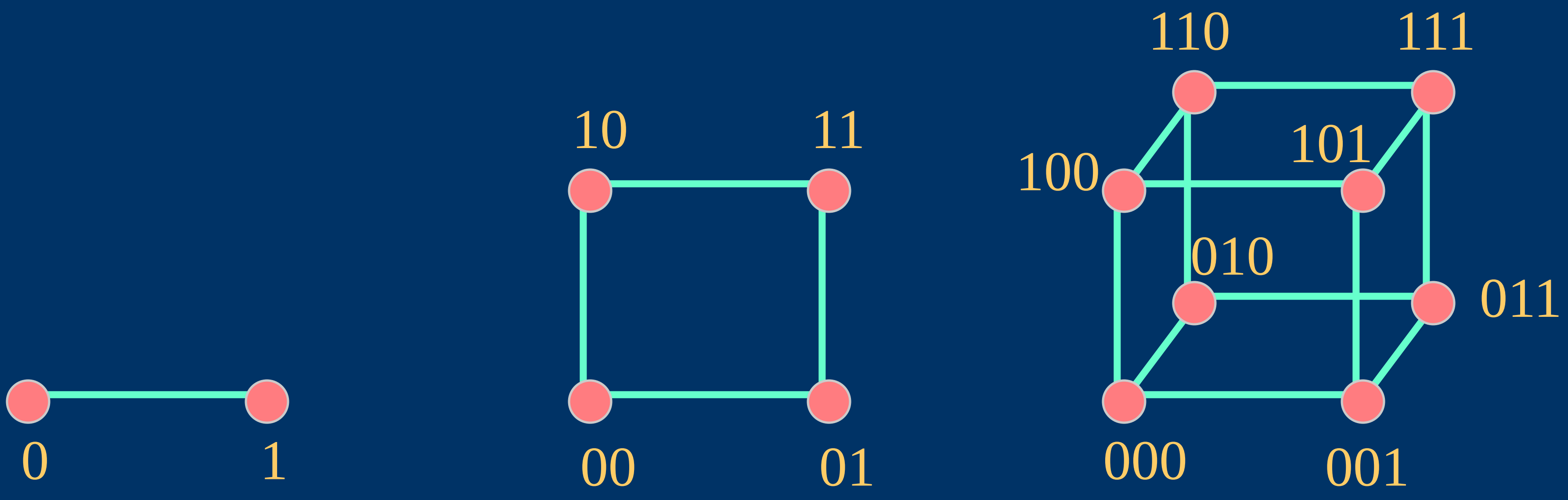
Graph Terminology (9.2) (cont.)

– n-cubes

They are denoted by Q_n, they are graphs that have vertices representing the 2^n bit strings of length n.

Two vertices are adjacent if and only if the bits strings that they represent differ in exactly one bit position

Graph Terminology (9.2) (cont.)



Q_1 Q_2 Q_3

The n-cube Q_n for n = 1, 2, and 3.

Graph Terminology (9.2) (cont.)

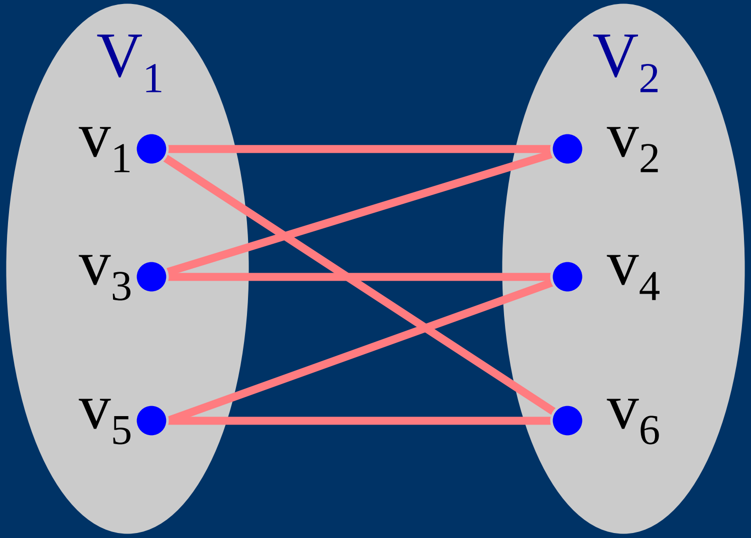
- Bipartite graph

  – Definition 5:

  A simple graph is called bipartite if its vertex set V can be partitioned into 2 disjoint sets V_1 and V_2 such that every edge in the graph connects a vertex in V_1 and a vertex in V_2 (so that no edge in G connects either 2 vertices in V_1 or 2 vertices in V_2).

Graph Terminology (9.2) (cont.)

– Example: C_6 is bipartite, since its vertex set can be partitioned into the 2 sets $V_1 = \{v_1, v_3, v_5\}$ and $V_2 = \{v_2, v_4, v_6\}$, and every edge of C_6 connects a vertex in V_1 and a vertex in V_2.



– Example: K_3 is not bipartite. Why?

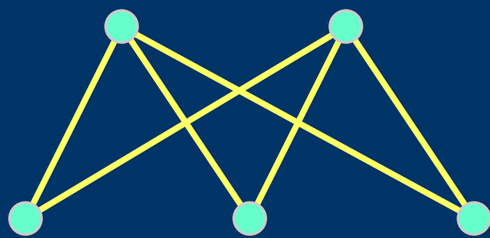
Graph Terminology (9.2) (cont.)

– Characterization of bipartite graph

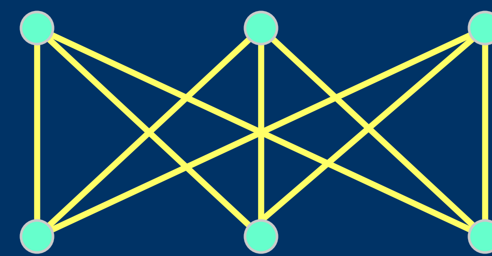
- A graph is bipartite if and only if it is possible to color the vertices of the graph with at most 2 colors so that no 2 adjacent vertices have the same color

- Example: Complete bipartite graphs: they are denoted by $K_{m,n}$. Their vertices set is partitioned into 2 subsets of m and n vertices, respectively. There is an edge between 2 vertices if and only if one vertex is in the first subset and the other vertex is in the second subset.

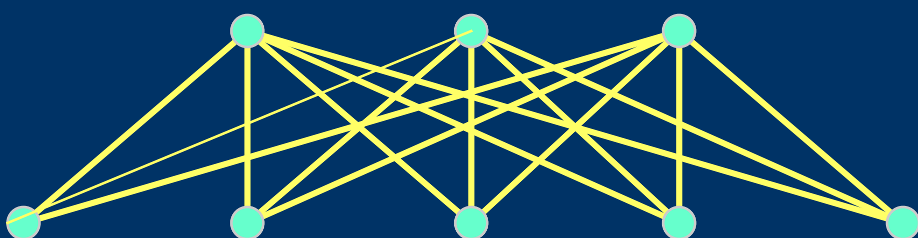
Graph Terminology (9.2) (cont.)



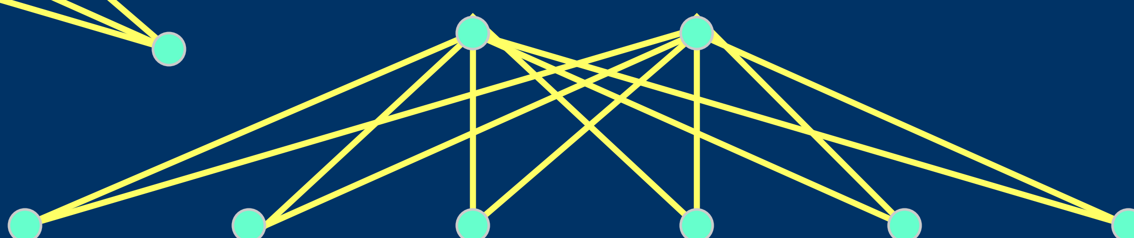
$K_{2,3}$



$K_{3,3}$



$K_{3,5}$



$K_{2,6}$

Some complete bipartite graphs

Graph Terminology (9.2) (cont.)

- Some applications of special types of graphs

  - Local area network
    Goal: connecting computers as well as peripheral devices in a building using a local area network topology

  - Some of these networks are based on a star topology, where all devices are connected to a central control device

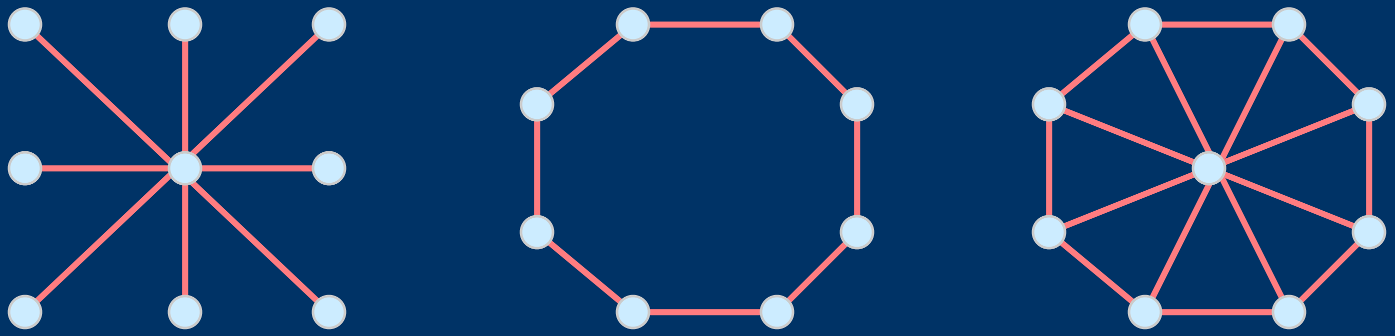
  - The star topology is equivalent to a $K_{1,n}$ complete bipartite graph

Graph Terminology (9.2) (cont.)

– Other local area networks use a ring topology $\Leftrightarrow$ C_n graphs

– Finally, the hybrid topology which is equivalent to a W_n graph is also used

Graph Terminology (9.2) (cont.)



(a) (b) (c)

Star, ring, and hybrid topologies for local area networks

Graph Terminology (9.2) (cont.)

– Interconnection networks for parallel processing

- Linear arrays for processor connection
- Mesh network (Markovian neighborhood)
- Hypercube interconnection (generalization of n-cubes)

Representing Graphs & Graph Isomorphism

- Introduction

  – Goal: Consists of choosing the most convenient representation of a graph

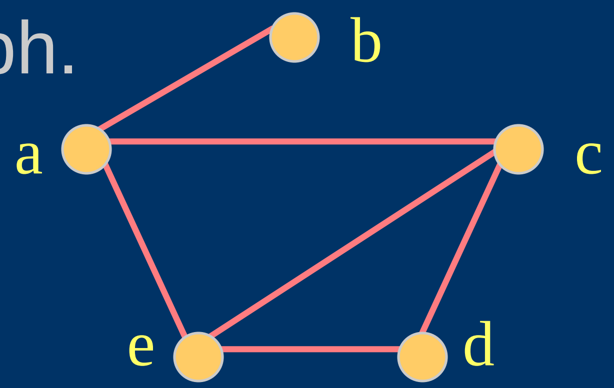
  – We need to determine whether 2 graphs are isomorphic, this problem is important in graph theory

Representing Graphs & Graph Isomorphism (9.3) (cont.)

- Representing Graph
  - List all the edges of the graph (no multiple edges)
  - Use adjacency list, which specifies the vertices that are adjacent to each vertex of the graph

Representing Graphs & Graph Isomorphism (9.3) (cont.)

– Example: Use adjacency lists to describe this simple graph.



Solution:

Vertex	Adjacent vertices
a	b, c, e
b	a
c	a, d, e
d	c, e
e	a, c, d

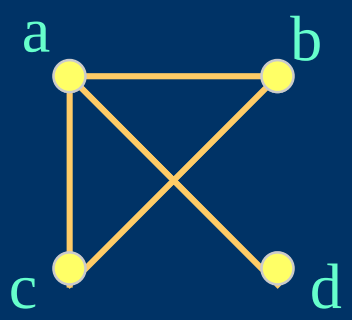
Representing Graphs & Graph Isomorphism (9.3) (cont.)

- Adjacency matrices
  - To simplify computation, graphs can be represented using matrices
    - Adjacency matrix
    - Incident matrix
  - The adjacency matrix is defined as A = [a_{ij}] such that

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

Representing Graphs & Graph Isomorphism (9.3) (cont.)

– Example: Use a adjacency matrix to represent this graph:



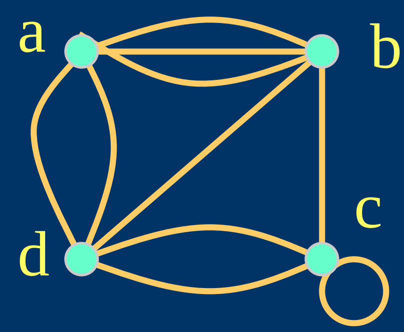
Solution: We order the vertices a, b, c, d. The matrix representing this graph is

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

Representing Graphs & Graph Isomorphism (9.3) (cont.)

– In case of pseudographs , the adjacency matrix is not a binary matrix but is formed of elements that represent the number of edges between 2 vertices

– Example: Use an adjacency matrix to represent this pseudograph:



Solution: The adjacency matrix using The ordering of vertices a, b, c, d is:

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

Representing Graphs & Graph Isomorphism (9.3) (cont.)

- Incidence matrices

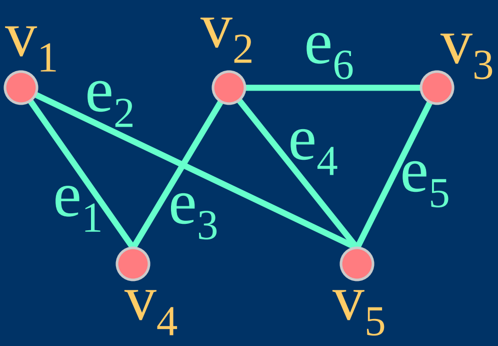
  – Let G = (V,E) be an undirected graph.

  – Incidence matrices are defined by the matrix M = $[m_{ij}]$ such that

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

Representing Graphs & Graph Isomorphism (9.3) (cont.)

– Example: Using an incidence matrix, represent the following undirected graph:



Solution:

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Representing Graphs & Graph Isomorphism (9.3) (cont.)

- Isomorphism of graphs
  - Goal: is it possible to draw 2 graphs in the same way?
  - In chemistry, different graph compounds can have the same molecular formula but can differ in structure
  - The graphs of these compounds cannot be drawn in the same way
  - Graphs having the same structure share common properties

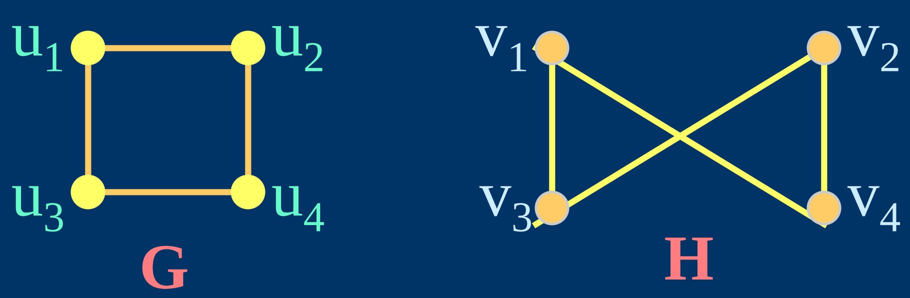
Representing Graphs & Graph Isomorphism (9.3) (cont.)

– Definition 1:

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.

Representing Graphs & Graph Isomorphism (9.3) (cont.)

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



Solution: The function f with $f(u_1) = v_1$, $f(u_2) = v_4$, $f(u_3) = v_3$, $f(u_4) = v_2$ is a one-to-one correspondence between V and W. To see that this correspondence preserves adjacency, 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, and 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 $f(u_3) = v_3$, and $f(u_1) = v_1$ and $f(u_4) = v_2$ are adjacent in H.

Connectivity (9.4)

- Goal: determination of paths within graphs
- Many problems can be modeled with paths formed by traveling along the edges of graphs
- Some examples of problems are:
  - Study the link between remote computers
  - Efficient planning of routes for mail delivery
  - Garbage pickup
  - Diagnostic in computer networks

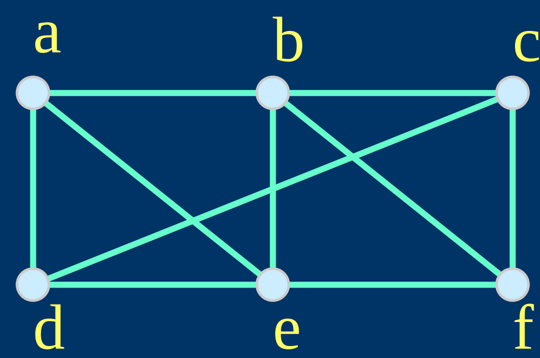
Connectivity (9.4) (cont.)

- Path
  - Definition 1:

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 $e_1, e_2, \ldots, e_n$ of G such that $f(e_1) = \{x_0, x_1\}$, $f(e_2) = \{x_1, x_2\}, \ldots, f(e_n) = \{x_{n-1}, x_n\}$, where $x_0 = u$ and $x_n = v$. When the graph is simple, we denote this path by its vertex sequence $x_0, x_1, \ldots, x_n$ (since listing these vertices uniquely determines the path). The path is a circuit if it begins and ends at the same vertex, that is, if u = v, and has length greater than zero. The path or circuit is said to pass through the vertices $x_1, x_2, \ldots, x_{n-1}$ or traverse the edges $e_1, e_2, \ldots, e_n$. A path or circuit is simple if it does not contain the same edge more than once.

Connectivity (9.4) (cont.)

– Example:



In this simple graph a, d, c, f, e is a simple path of length 4, since {a,d}, {d,c}, {c,f}, and {f,e} are all edges. However, d, e, c, a is not a path, since {e,c} is not an edge.
Note that b, c, f, e, b is a circuit of length 4 since {b,c}, {c,f}, {f,e}, and {e,b} are edges, and this path begins and ends at b.
The path a, b, e,d,a,b, which is of length 5, is not simple since it contains the edge {a,b} twice.

Connectivity (9.4) (cont.)

– Definition 2:

Let n be a nonnegative integer and G a directed multigraph. A path of length n from u to v in G is a sequence of edges $e_1, e_2, \ldots, e_n$ of G such that $f(e_1) = (x_0, x_1), f(e_2) = (x_1, x_2), \ldots, f(e_n) = (x_{n-1}, x_n)$, where $x_0 = u$ and $x_n = v$. When there are no multiple edges in the directed graph, this path is denoted by its vertex sequence $x_0, x_1, \ldots, x_n$. A path of length greater than zero that begins and ends at the same vertex is called a circuit or cycle. A path or circuit is called simple it it does not contain the same edge more than once.

Connectivity (9.4) (cont.)

- Connectedness in undirected graphs
  - Question asked:
    When does a computer network have the property that every pair of computers can share information, if message can be sent through one or more intermediate computers?
  - This question is equivalent to:
    When is there always a path between 2 vertices in the graph?

Connectivity (9.4) (cont.)

– Definition 3:

An undirected graph is called connected if there is a path between every pair of distinct vertices of the graph

– Theorem 1:

There is a simple path between every pair of distinct vertices of a connected undirected graph

Proof: Exercise!

Connectivity (9.4) (cont.)

- A graph that is not connected is the union of two or more connected subgraphs, each pair of which has no vertex in common.

- These disjoint connected subgraphs are called the connected components of the graph

Connectivity (9.4) (cont.)

- Connected in directed graphs
  - Definition 4:

    A directed graph is strongly connected if there is a path from a to b and from b to a whenever a and b are vertices in the graph

  - Definition 5:

    A directed graph is weakly connected if there is a path between every 2 vertices in the underlying undirected graph

Connectivity (9.4) (cont.)

– A directed graph is weakly connected $\Leftrightarrow$ there is always a path between 2 vertices when the directions of the edges are ignored

– Strongly connected $\Rightarrow$ weakly connected directed graph

Connectivity (9.4) (cont.)

– Example: Are the directed graphs G and H strongly connected?

Solution: G is strongly connected because there is a path between any 2 vertices in this directed graph. Hence, G is also weakly connected.

The graph H is not strongly connected. There is no directed path from a to b in this graph. However, H is weakly connected, since there is a path between any 2 vertices in their underlying undirected graph of H.

Connectivity (9.4) (cont.)

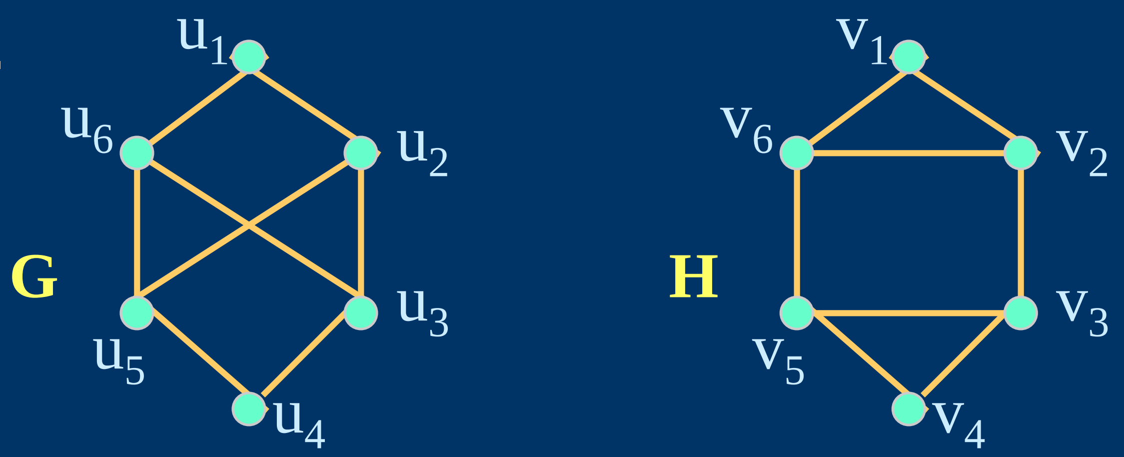
- Paths & isomorphism

  – Paths and circuits can help determine whether 2 graphs are isometric

  – The existence of a simple circuit (or cycle) of a particular length is a useful invariant to show that 2 graphs are not isomorphic

Connectivity (9.4) (cont.)

– Example: Determine whether the graph G and H are isomorphic.



Solution: Both G and H have 6 vertices and 8 edges. Each has 4 vertices of degree 3, and two vertices of degree 2. However, H has a simple circuit of length 3, namely, v_1, v_2, v_6, v_1 whereas G has no simple circuit of length 3, as can be determined by inspection (all simple circuits in G have length at least four). Since the existence of a simple circuit of length 3 is an isomorphic invariant, G and H are not isomorphic.

Connectivity (9.4) (cont.)

- Counting paths between vertices

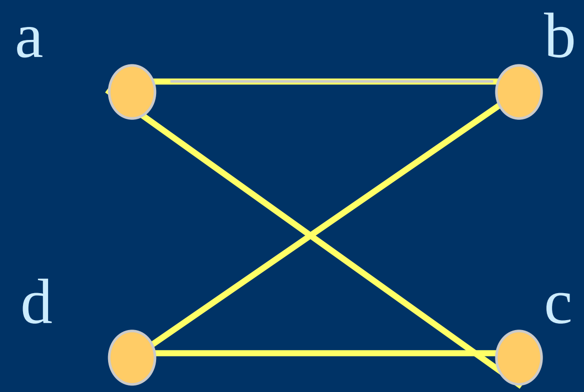
  - Theorem 2:

    Let G be a graph with adjacency matrix A with respect to the ordering $v_1, v_2, \ldots, v_n$ (with directed or undirected edges, with multiple edges and loops allowed). The number of different paths of length r from v_i to v_j, where r is a positive integer is equals to the (i, j)th entry of A^r.

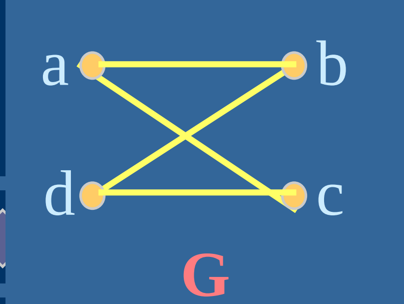
Proof: Exercise!

Connectivity (9.4) (cont.)

– Example: How many paths of length 4 are there from a to d in the simple graph G?



G



Connectivity (9.4) (cont.)

Solution: The adjacency matrix of G (ordering the vertices as a, b, c, d) is

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

Hence, the number of paths of length 4 from a to d is the (1,4)th entry of A^4. Since

$$A^4 = \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix}.$$

there are exactly 8 paths of length 4 from a to d. By inspection of the graph, we see that a, b, a, b, d; a, b, a, c, d; a, b, d, b, d; a, b, d, c, d; a, c, a, b, d; a, c, a, c, d; a, c, d, b, d; and a, c, d, c, d are the 8 paths from a to d.