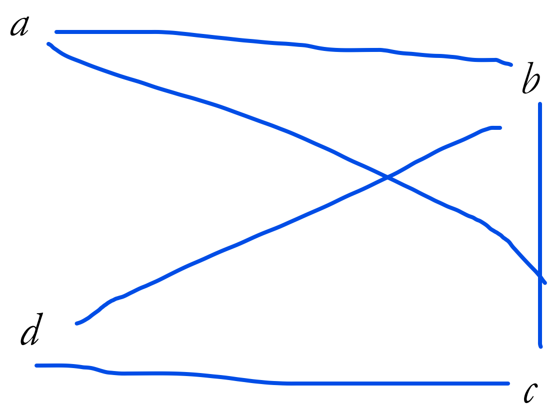
Discrete Structures Spring 2024 – Week 10

Graphs

Lecture 1

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: A graph with four vertices and five edges



Blue = edges

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.

a b

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

Directed Graphs

- **Definition:** A *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. [a][b]
- Example: Directed graph with three vertices and four edges

- A *directed multigraph* may have multiple directed edges. When there are *m* [c]

  directed edges from the vertex *u* to the vertex *v*, we say that (*u*,*v*) is an edge of multiplicity m.

- Example: Directed multigraph the multiplicity of (a,b) is 1 and the multiplicity of (b,c) is 2.

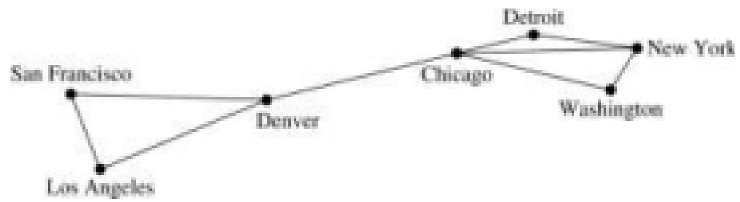
a b

c

Graph Models: Computer Networks

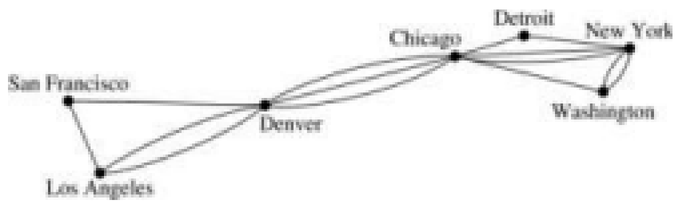
- When we build a graph model, we use the appropriate type of graph to capture the important features of the application.
- To model a computer network where we are only concerned whether two data centers are connected by a communications link, we use a **simple**

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.

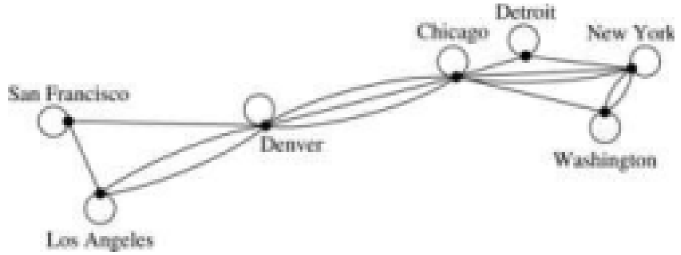


Graph Models: Computer Networks

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

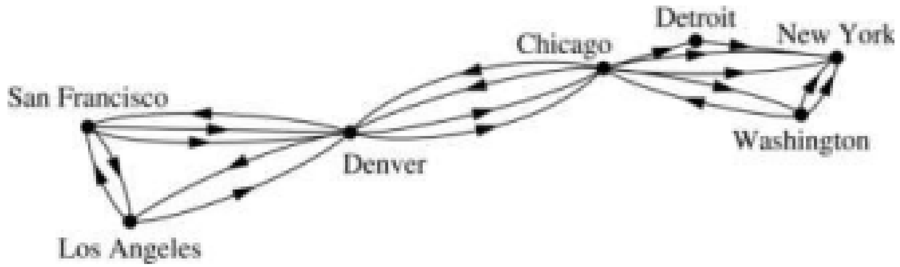


- To model a computer network with diagnostic links at data centers, we use a *pseudograph*, as loops are needed.



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 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

• Graph theory can be used in models of:

- Social networks
- Communications networks
- Information networks
- Software design
- Transportation networks
- Biological networks

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.

Graph Models: Social Networks

- 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.)
  - **Influence graphs** – directed graphs where there is an edge from one person to another if the first person can influence the second person
- **Collaboration graphs** – undirected graphs where two people are connected if they collaborate in a specific way

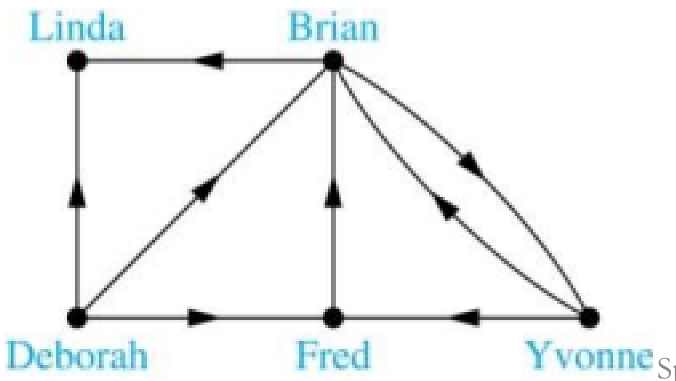
Graph Models: Social Networks (continued)

- **Example:** A friendship graph

where two people are connected if they are Facebook friends.

•

• **Example:** An influence graph



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.

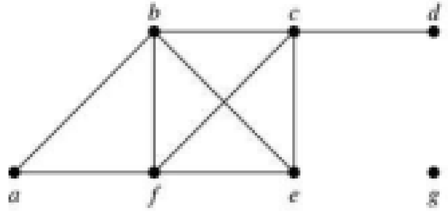
- **Definition 2**. The set of all neighbors of a vertex v of $G = (V, E)$, denoted by $N(v)$, is called the *neighborhood* of v. If A is a subset of V, we denote by $N(A)$ the set of all vertices in G that are adjacent to at least one vertex in A. So, $N(A) = \bigcup_{v \in A} N(v)$.

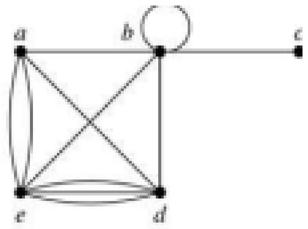
Basic Terminology

- **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





graph G?

- **Solution**

G: $\deg(a) = 2$, $\deg(b) = \deg(c) = \deg(f) = 4$, $\deg(d) = 1$,

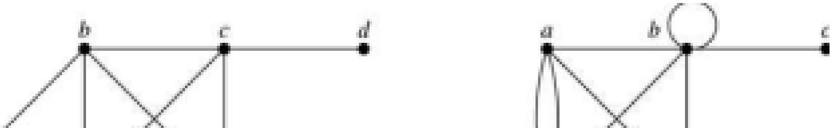
$\deg(e) = 3$, $\deg(g) = 0$.

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

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

Degrees and Neighborhoods of Vertices

- **Example**: What are the degrees and



neighborhoods of the vertices in the graph H?

- **Solution**

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 – Degrees of Vertices

- **Theorem 1 (Handshaking Theorem)**: If $G = (V,E)$ is an undirected graph with m edges, then $2m = \sum_{v \in V} \deg(v)$
- **Proof**:
- Each edge contributes twice to the degree count of all vertices. Hence, both the left-hand and right-hand sides of this equation equal twice the number of edges.
- *Think about the graph where vertices represent the people at a party and an edge connects two people who have shaken hands.*

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 $m = 30$

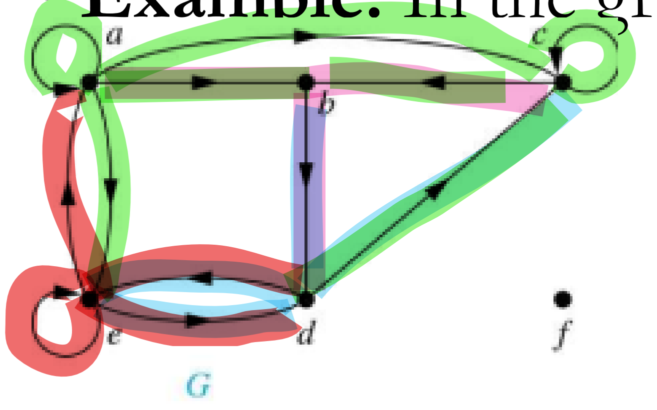
Directed Graphs

- **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

- **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

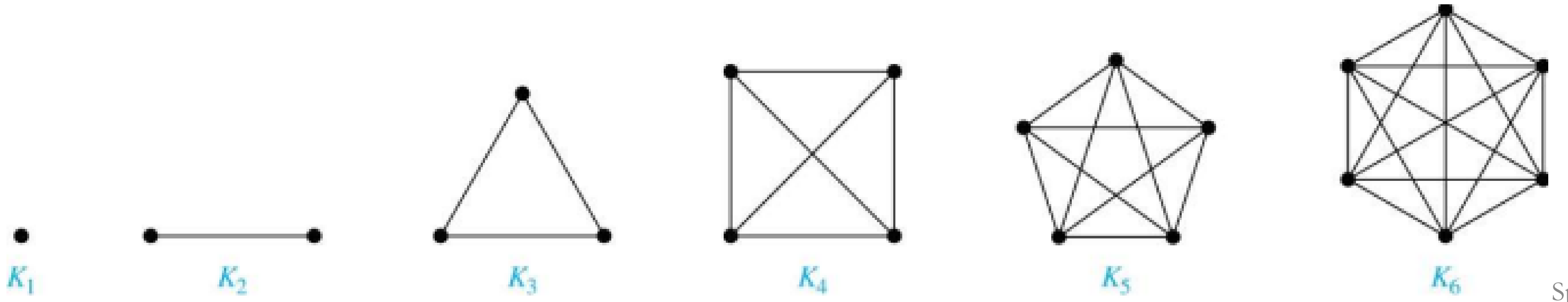
- **Theorem 3**: Let $G = (V, E)$ be a graph with directed edges. Then:

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

- **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

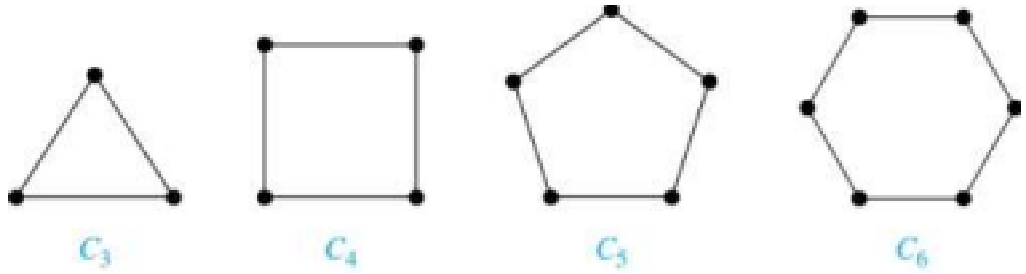
- A *complete graph on n vertices*, denoted by K_n, is the simple graph that contains exactly one edge between each pair of distinct vertices.



Special Types of Simple Graphs:

Cycles

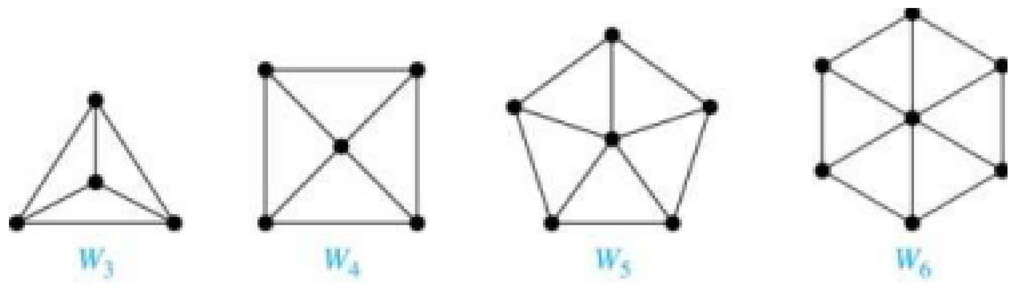
- A *cycle* C_n for $n \geq 3$ consists of n vertices $v_1, v_2, \cdots, v_n$, and edges $\{v_1, v_2\}$, $\{v_2, v_3\}, \cdots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$.



Special Types of Simple Graphs:

Wheels

- 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.



Bipartite Graphs

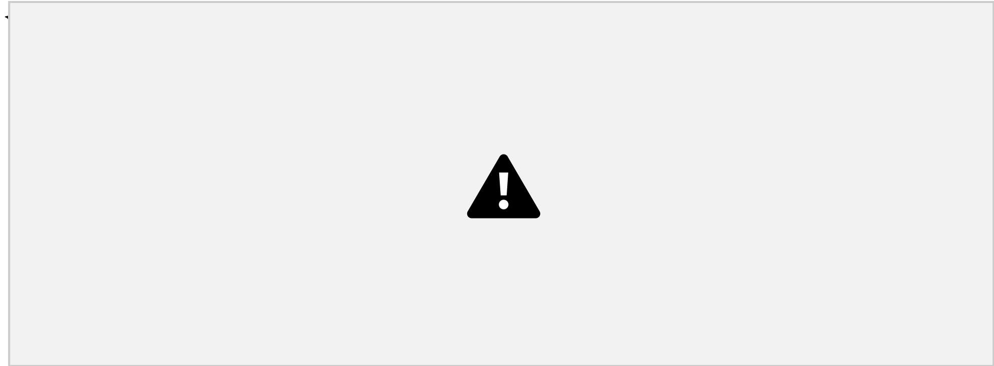
Definition: A simple graph G is bipartite if V can be partitioned into two disjoint subsets V_1 and V_2 such that every edge connects a vertex in V_1 and a vertex in V_2. In other words, there are no edges which connect two vertices in V_1 or in V_2.

Bipartite Graphs

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.

H is not bipartite

G is



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.

Representing Graphs

Lecture 2

Representing Graphs

1. Adjacency List
2. Adjacency Matrices
3. Incidence Matrices

1. 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.

1. Adjacency Lists

Example

1. Adjacency Lists

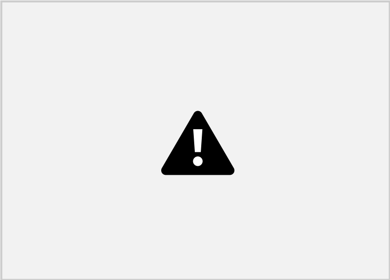
Example

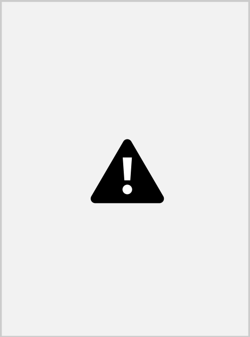
2. Adjacency Matrices

- **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

2. Adjacency Matrices





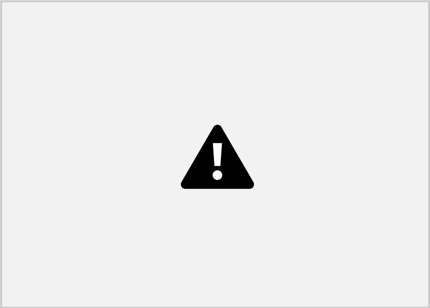
- Example

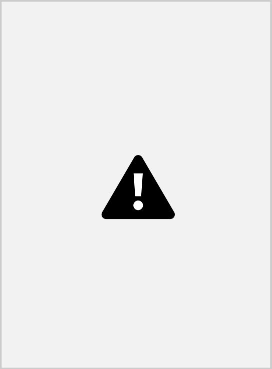
The ordering of vertices is a, b, c, d.

- **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

2. Adjacency Matrices

- Exercise



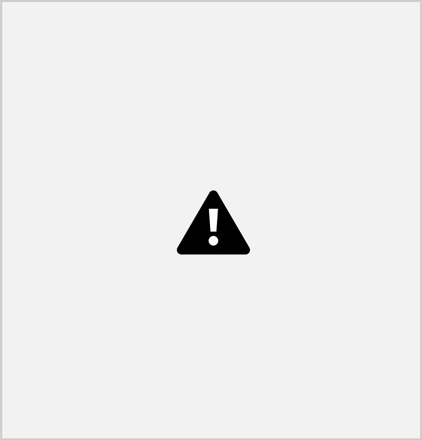


2. Adjacency Matrices

- Adjacency matrices can also be used to represent graphs with loops and multiple edges.
- A loop at the vertex v_i is represented by a 1 at the (i, j)th position of the matrix.
- When multiple edges connect the same pair of vertices v_i and v_j, (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.

2. Adjacency Matrices

- Example: We give the adjacency matrix of the pseudograph shown here using the ordering of vertices a, b, c, d.



2. Adjacency Matrices

- Adjacency matrices can also be used to represent directed graphs. The matrix for a directed graph $G = (V, E)$ has a $\mathbf{1}$ 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

2. Adjacency Matrices

- 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 multi-graphs, the value of a_{ij} is the number of edges connecting v_i to v_j.

2. Adjacency Matrices

- 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.

3. 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

3. Incidence Matrices

- **Example**: Simple Graph and Incidence Matrix

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.

3. Incidence Matrices

- **Exercise:** Pseudograph and Incidence Matrix

Euler Path and Circuit

Lecture 3

Euler Paths and Circuits

- 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.

The 7 Bridges of Königsberg

- People wondered whether it was possible to follow a path that crosses each bridge exactly once and returns to the starting point.

Euler Paths and Circuits

- 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.

Multigraph Model of the Bridges of Königsberg

Path and Circuit

- A **path** is a sequence of vertices with the property that each vertex in the sequence is adjacent to the vertex next to it.
- A path that does not repeat vertices is called a simple

path. • A **circuit** is path that begins and ends at the same

vertex.

Euler Paths and Circuits

- An *Euler path* in G is a simple path containing every edge of G. **Or** *Euler path* in G is a path that includes every edge of a G exactly once.

- An *Euler circuit* in a graph G is a simple circuit containing every edge of G. **Or** *Euler circuit* is a circuit that includes each edge exactly once

- An Euler circuit is always an Euler path, but an Euler path may not be an Euler circuit.

Euler Paths and Circuits

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.

Euler Theorem

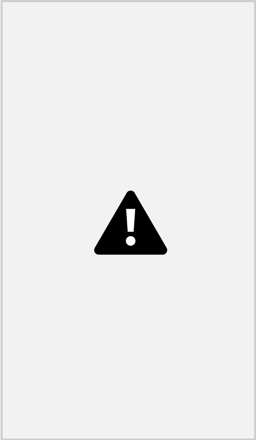
1. A connected multi-graph with at least two vertices has an Euler circuit if and only if each of its vertices has an even degree

2. The graph has an Euler path if and only if it has exactly two vertices of odd degree.

Euler Theorem

- **Example**:
- Two of the vertices in the multi-graph model of the Königsberg bridge

problem have odd degree. Hence, there is no Euler circuit in this multi graph and it is impossible to start at a given point, cross each bridge exactly once, and return to the starting point.



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.

Euler Paths and Circuits

- **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.

Euler Paths and Circuits

- **Example:**

- 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*.

Euler Paths and Circuits

- **Example:**

- 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 Path and Circuit

Lecture 4

Hamilton Paths and Circuits

- Euler paths and circuits contained every edge only once. Now we look at paths and circuits that contain every vertex exactly once.
- **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 $x_0, x_1, ..., x_{n-1}, x_n$ 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.

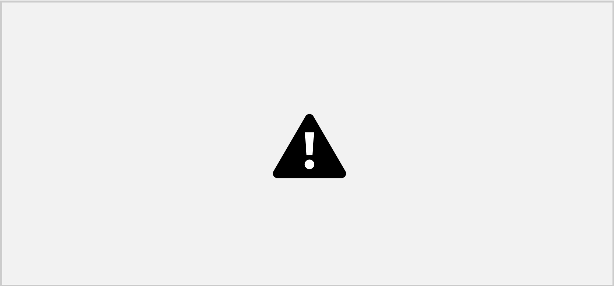
Hamilton Paths and Circuits

• 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.

William Rowan Hamilton (1805- 1865)



Hamilton Path

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 Hamiltonian 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.