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 (*not currently included in overheads*)
- Planar Graphs (*not currently included in overheads*)
- Graph Coloring (*not currently included in overheads*)

Connectivity

Section 10.4

Section Summary

- Paths
- Connectedness in Undirected Graphs
- Vertex Connectivity and Edge Connectivity (*not currently included in overheads*)
- Connectedness in Directed Graphs
- Paths and Isomorphism (*not currently included in overheads*)
- 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.

This terminology is readily extended to directed graphs. (*see text*)

Some Remarks

Remark 1:

The word "**simple**" is overloaded. Don't confuse a simple undirected graph with a simple path. There can be a simple path in a non-simple graph, and a non-simple path in asimple graph.

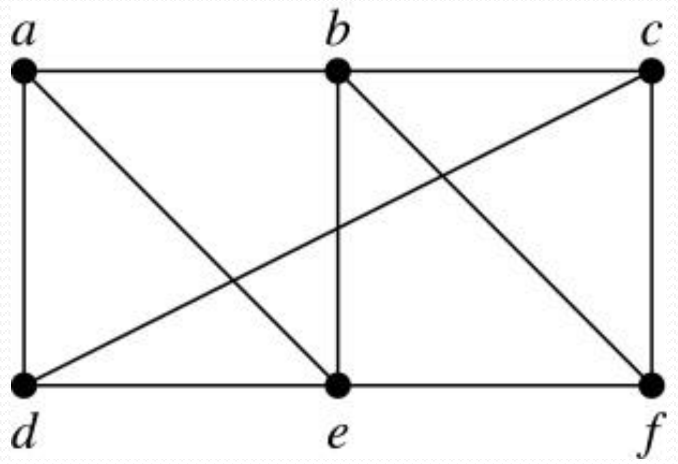
Remark 2:

The terms "**path**" and "**simple path**" used in Rosen's book are not entirely standard. Other books use the terms "*walk*" and "*trail*" to denote "path" and "simple path", respectively.

Paths (*continued*)

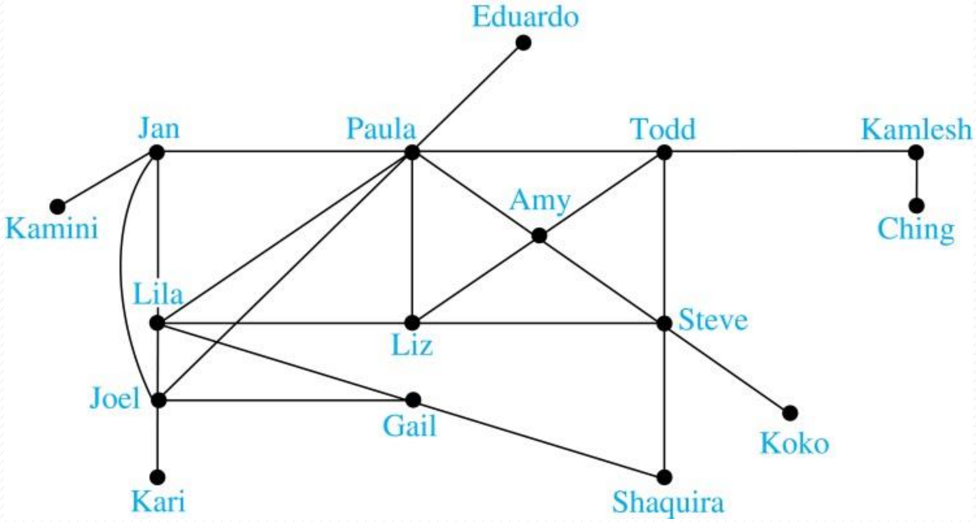
Example: In the simple graph here:

- *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.



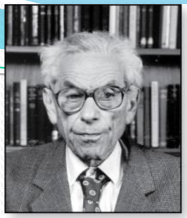
Degrees of Separation

Example: ***Paths in Acquaintanceship Graphs***. In an acquaintanceship graph there is a path between two people if there is a chain of people linking these people, where two people adjacent in the chain know one another. In this graph there is a chain of six people linking Kamini and Ching.



Some have speculated that almost every pair of people in the world are linked by a small chain of no more than six, or maybe even, five people. The play *Six Degrees of Separation* by John Guare is based on this notion.

Erdős numbers



Paul Erdős

TABLE 1 The Number of Mathematicians with a Given Erdős Number (as of early 2006).

Erdős Number	*Number of People*
0	1
1	504
2	6,593
3	33,605
4	83,642
5	87,760
6	40,014
7	11,591
8	3,146
9	819
10	244
11	68
12	23
13	5

Example: *Erdős numbers*.
In a collaboration graph, two people *a* and *b* are connected by a path when there is a sequence of people starting with *a* and ending with *b* such that the endpoints of each edge in the path are people who have collaborated.

- In the **academic collaboration graph of people** who have written papers in mathematics, the *Erdős number* of a person *m* is the length of the shortest path between *m* and the prolific mathematician Paul Erdős.
- To learn more about Erdős numbers, visit

http://www.ams.org/mathscinet/collaborationDistance.html

Bacon Numbrers

- In the Hollywood graph, two actors *a* and *b* are linked when there is a chain of actors linking *a* and *b*, where every two actors adjacent in the chain have acted in the same movie.
- The *Bacon number* of an actor *c* is defined to be the length of the shortest path connecting *c* and the well-known actor Kevin Bacon. (Note that we can define a similar number by replacing Kevin Bacon by a different actor.)
- The *oracle of Bacon* web site http://oracleofbacon.org/how.php provides a tool for finding Bacon numbers.

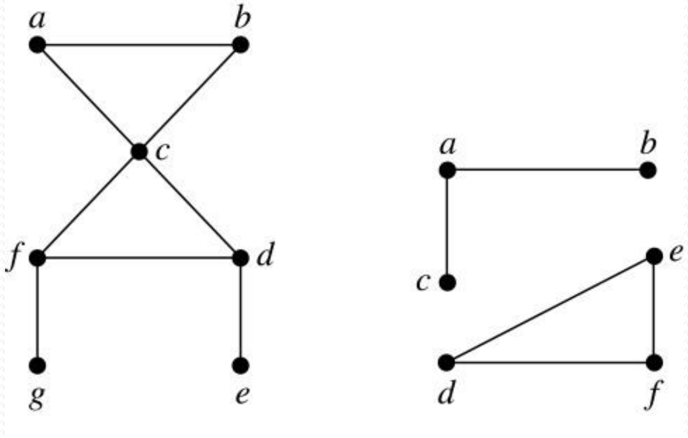
TABLE 2 The Number of Actors with a Given Bacon Number (as of early 2011).

Bacon Number	*Number of People*
0	1
1	2,367
2	242,407
3	785,389
4	200,602
5	14,048
6	1,277
7	114
8	16

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.

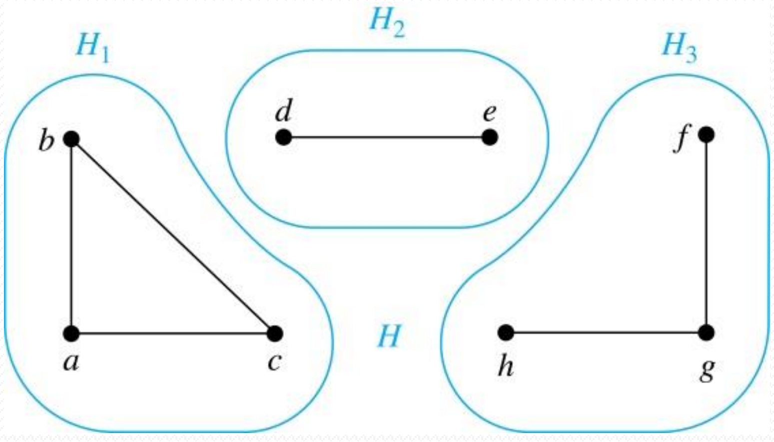


G_1 G_2

Connected Components

Definition: A *connected component* of a graph G is a connected subgraph of G that is not a proper subgraph of another connected subgraph of G. A graph G that is not connected has two or more connected components that are disjoint and have G as their union.

Example: The graph H is the union of three disjoint subgraphs H_1, H_2, and H_3. These three subgraphs are the connected components of H.



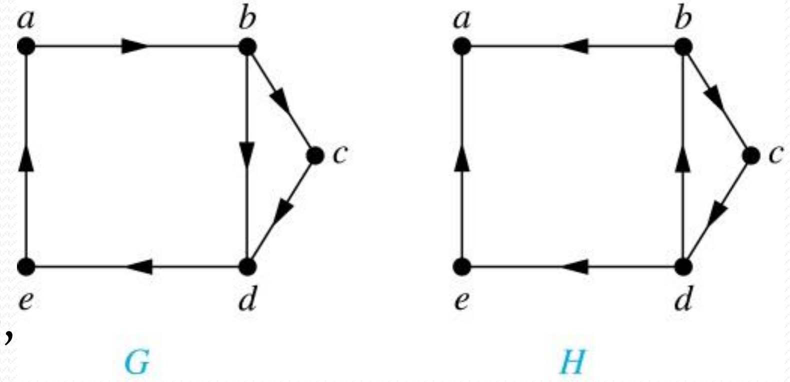
Connectedness in Directed Graphs

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

Definition: A directed graph is *weakly connected* if there is a path between every two vertices in the underlying undirected graph, which is the undirected graph obtained by ignoring the directions of the edges of the directed graph.

Connectedness in Directed Graphs (*continued*)

Example: G is strongly connected because there is a path between any two vertices in the directed graph. Hence, G is also weakly connected. The graph H is not strongly connected, since there is no directed path from a to b, but it is weakly connected.



Definition: The subgraphs of a directed graph G that are strongly connected but not contained in larger strongly connected subgraphs, that is, the maximal strongly connected subgraphs, are called the *strongly connected components* or *strong components* of G.

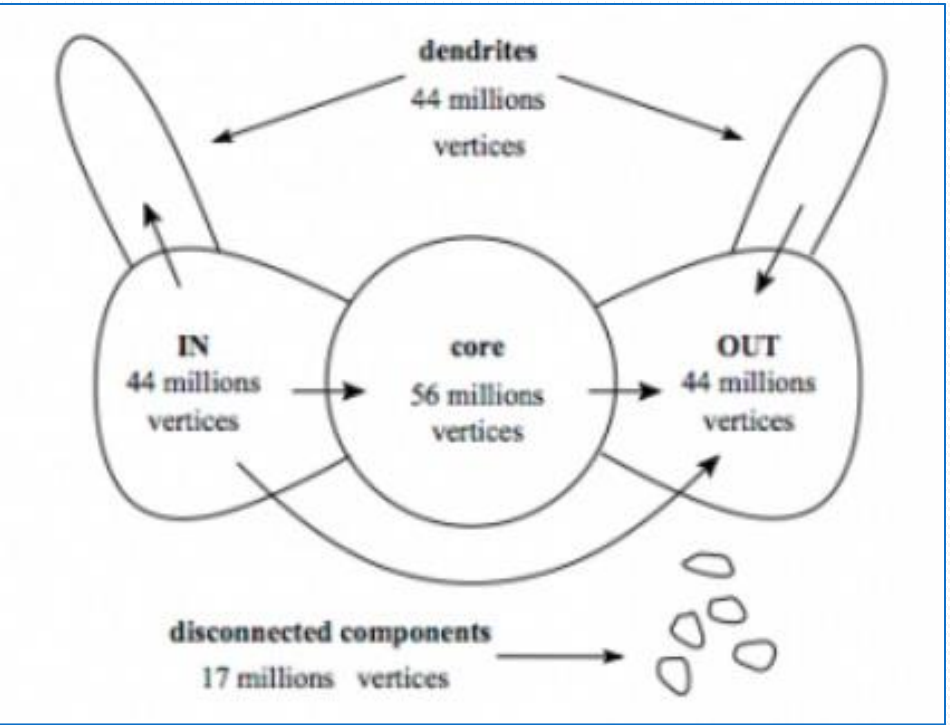
Example (*continued*): The graph H has three strongly connected components, consisting of the vertex a; the vertex e; and the subgraph consisting of the vertices b, c, d and edges (b,c), (c,d), and (d,b).

The Connected Components of the Web Graph (opt.)

Recall that at any particular instant the web graph provides a snapshot of the web, where vertices represent **web pages and edges represent links.**

The underlying undirected graph of this Web graph has a connected component that includes approximately 90% of the vertices.

There is a ***giant strongly connected component (GSCC)*** consisting of more than 53 million vertices. A Web page in this component can be reached by following links starting in any other page of the component.



There are three other categories of pages with each having about 44 million vertices: 1) pages that can be reached from a page in the GSCC, but do not link back., 2) that link back to the GSCC, but can not be reached by following links from pages in the GSCC, 3) pages that cannot reach pages in the GSCC and can not be reached from pages in the GSCC.

Counting Paths (or walks) between Vertices

We can use the adjacency matrix of a graph to find the number of paths (or walks) between two vertices in the graph.

Theorem: Let G be a graph with adjacency matrix **A** with respect to the ordering $v_1, \ldots, v_n$ of vertices (with directed or undirected edges, multiple edges and loops allowed). The **number of different paths (or walks) of length *r* from v_i to v_j**, where $r > 0$ is a positive integer, equals the (i,j)th entry of $\mathbf{A}^r$.

Proof by mathematical induction: (optional, look after Counting Lecture)

Basis Step: By definition of the adjacency matrix, the number of paths from v_i to v_j of length 1 is the (i,j)th entry of $\mathbf{A}$.

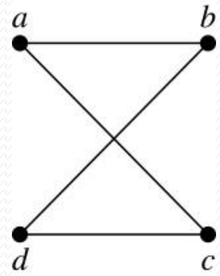
For the inductive hypothesis, we assume that that the (i,j)th entry of $\mathbf{A}^r$ is the number of different paths of length r from v_i to v_j.

Inductive Step: Because $\mathbf{A}^{r+1} = \mathbf{A}^r\,\mathbf{A}$, the (i,j)th entry of $\mathbf{A}^{r+1}$ equals $b_{i1}a_{1j} + b_{i2}a_{2j} + \cdots + b_{in}a_{nj}$, where b_{ik} is the $(i,k)th$ entry of $\mathbf{A}^r$. By the inductive hypothesis, b_{ik} is the number of paths of length r from v_i to v_k.

A path of length $r + 1$ from v_i to v_j is made up of a path of length r from v_i to some v_k, and an edge from v_k to v_j. By the product rule for counting, the number of such paths is the product of the number of paths of length r from v_i to v_k (i.e., b_{ik}) and the number of edges from from v_k to v_j (i.e, a_{kj}). The sum over all possible intermediate vertices v_k *is* $b_{i1}a_{1j} + b_{i2}a_{2j} + \cdots + b_{in}a_{nj}$.

Counting Paths between Vertices (*continued*)

Example: How many paths of length four are there from a to d in the graph G.



$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$ *adjacency matrix of G*

Solution: The adjacency matrix of G (ordering the vertices as a, b, c, d) is given above. Hence the number of paths of length four from a to d is the (1, 4)th entry of $\mathbf{A}^4$. The eight paths are as:

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

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	a, c, d, c, d

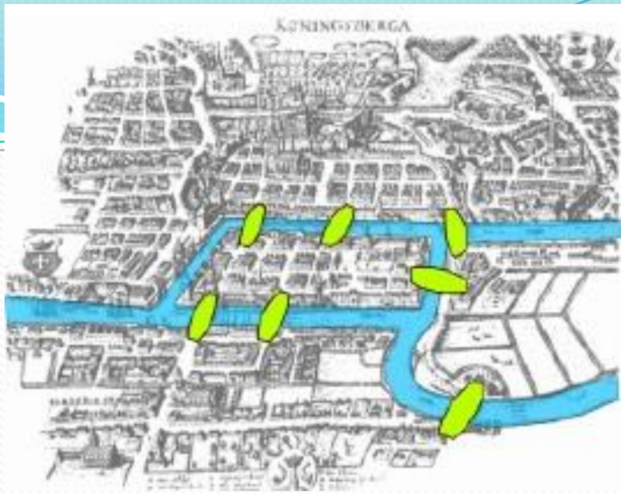
Euler and Hamiltonian Graphs

Section 10.5

Section Summary

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

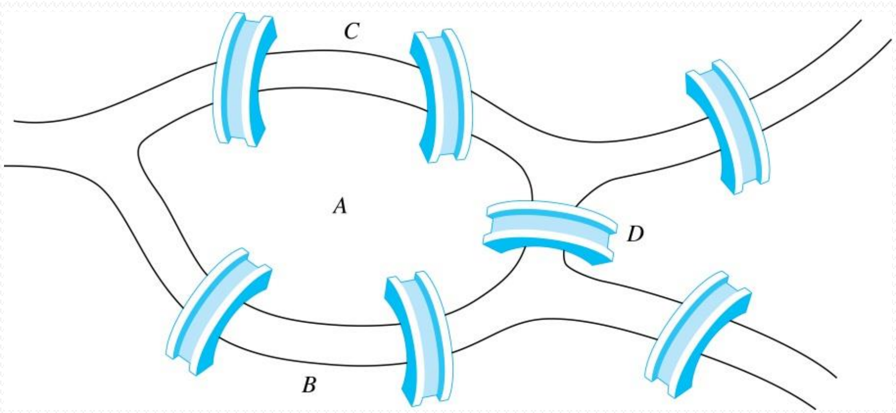
Euler Paths and Circuits



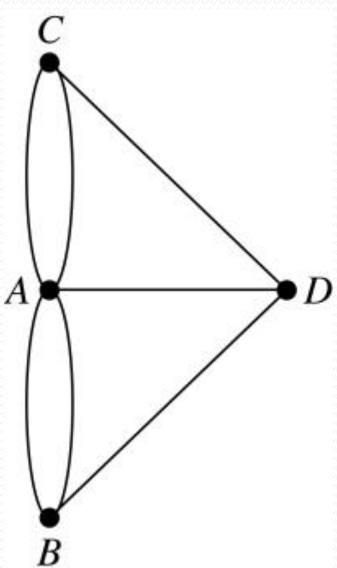
- The town of Königsberg, Prussia (now Kalingrad,

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

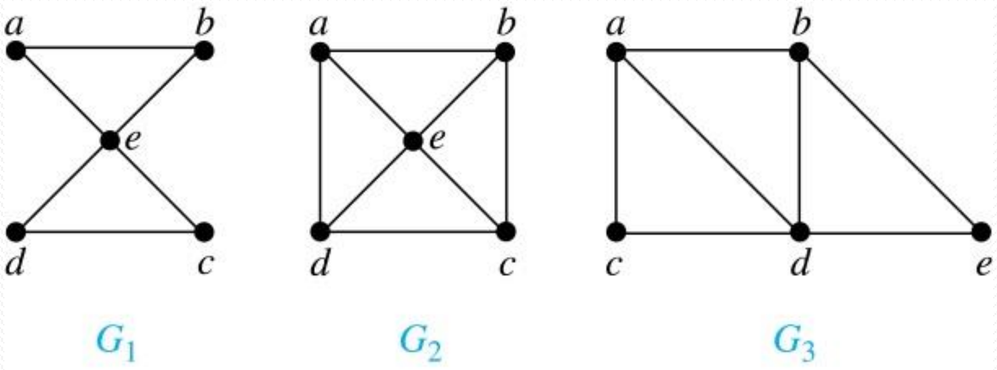


Multigraph Model of the Bridges of Königsberg

Euler Paths and Circuits (*continued*)

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:

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.

Theorem: A connected graph G has an Euler circuit **iff** each vertex of G has even degree.

Proof : [The “only if” case]

If the graph has an Euler circuit, then when we walk along the edges according to this circuit, each vertex must be entered and exited the same number of times.
Thus, the degree of each vertex must be even.

Theorem: A connected graph G has an Euler circuit **iff** each vertex of G has even degree.

Proof : [The "if" case]

Algorithm 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 **// (Note that after each iteration, the degrees of all vertices are still even.)**

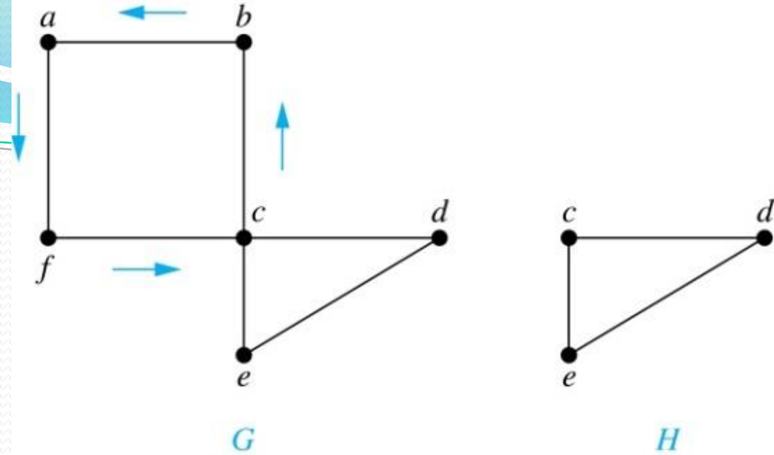
subcircuit := a circuit in *H* beginning at a vertex in *H* that also is an endpoint of an edge in circuit.

H := *H* with edges of *subciruit* and all isolated vertices removed

circuit := *circuit* with *subcircuit* inserted at the appropriate vertex.

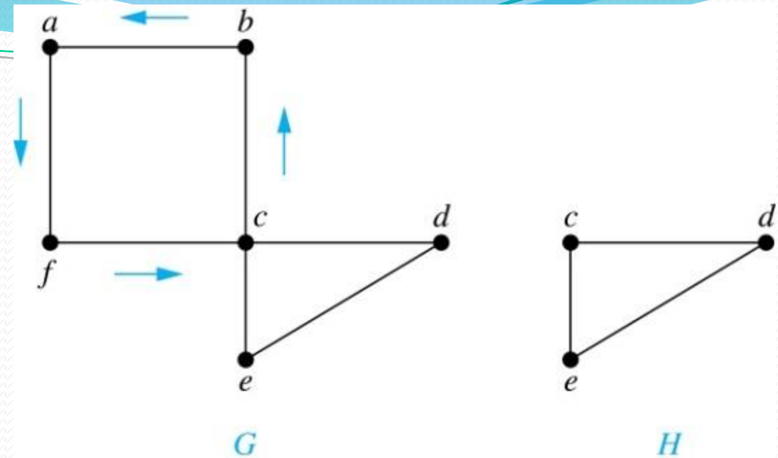
return *circuit*{*circuit* is an Euler circuit}

Necessary 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 (see the Algorithm)



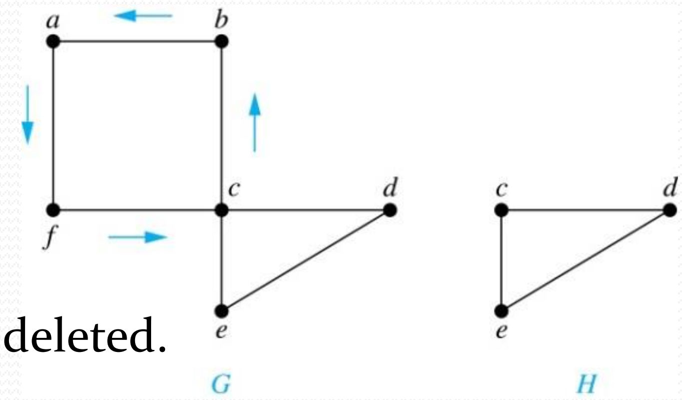
Suppose that G is a connected multigraph with ≥ 2 vertices, all of even degree. Let $x_0 = a$ be a vertex of even degree. Choose an edge $\{x_0, x_1\}$ incident with a and proceed to build a simple path $\{x_0, x_1\}, \{x_1, x_2\}, \ldots, \{x_{n-1}, x_n\}$ by adding edges one by one until another edge 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 (*continued*)



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*.

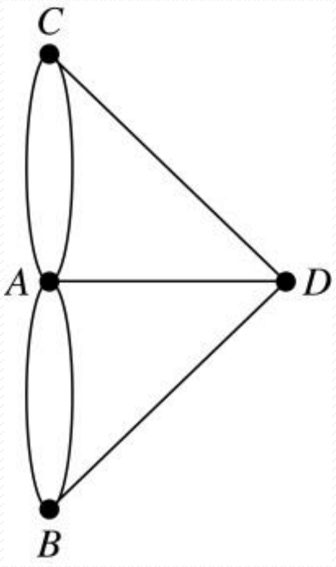
Continue this process until all edges have been used. This produces an Euler circuit. Since every edge is included and no edge is included more than once.

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

Theorem: A connected graph G has an Euler path if and only if it has exactly two vertices of odd degree.

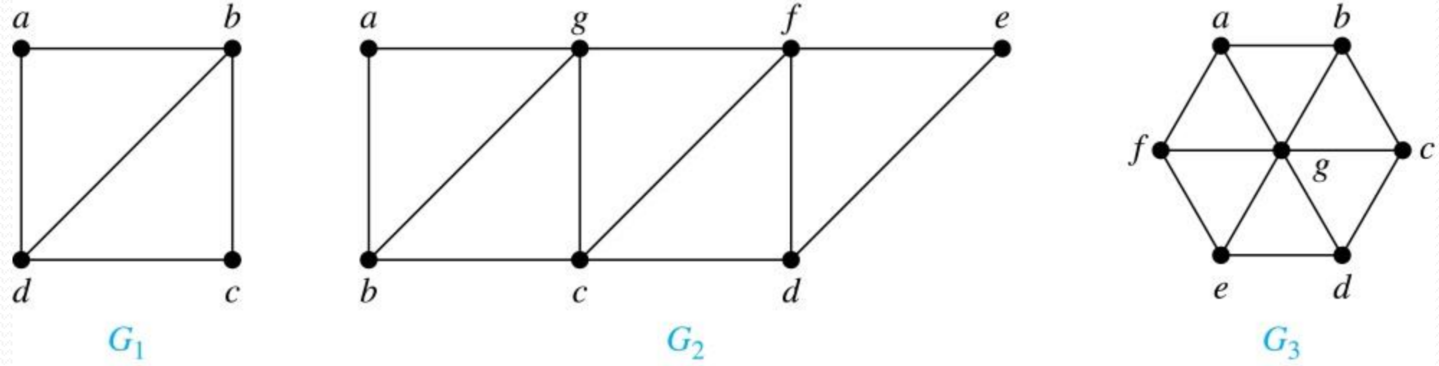
Proof :

Repeat the proof for Euler circuit after adding an edge between the vertices with odd degrees.



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

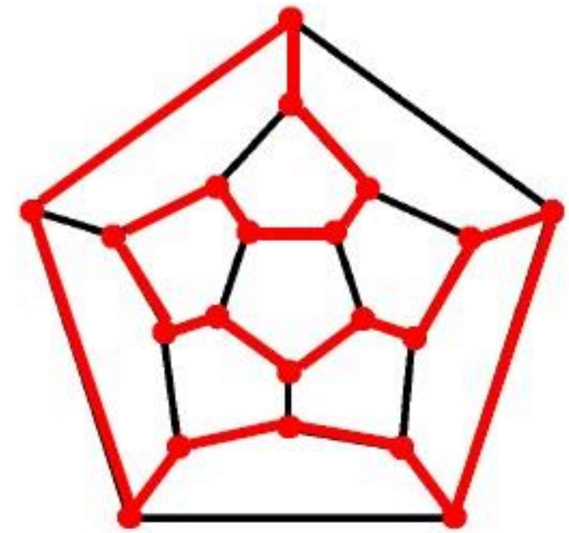
- Euler paths and circuits can be used to solve many practical problems such as finding a path or circuit that traverses each
  - street in a neighborhood,
  - road in a transportation network,
  - connection in a utility grid,
  - link in a communications network.
- Other applications are found in the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

William Rowan Hamilton
(1805- 1865)

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 two graphs below are isomorphic.



Hamilton Paths and Circuits

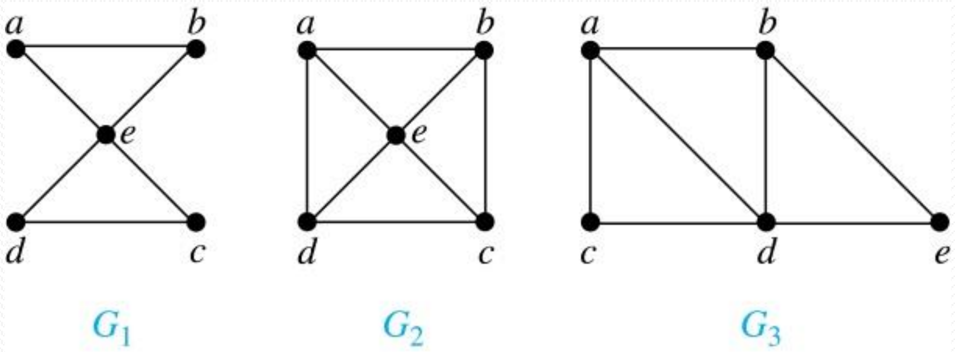
Definition: A simple path in a graph G that passes **through every vertex exactly once** is called a **Hamilton path,** and a simple circuit in a graph G that passes through every vertex exactly once is called a **Hamilton circuit .**

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 $\boldsymbol{V = \{x_0, x_1, \ldots, x_{n-1}, x_n\}}$ and $\boldsymbol{x_i \neq x_j}$ **for** $\boldsymbol{0 \leq i < j \leq n}$, 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.

Remark: *In many resources, Hamilton circuit is called Hamilton cycle.*

Hamilton Paths and Circuits (*continued*)

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, but has a Hamilton path: *a, b, e, d, c.*
- G_2 has a Hamilton circuit: *a, e, b, c, d, a*.
- G_3 has a Hamilton circuit: a, b, e, d, c, a.

Sufficient Conditions for Hamilton Circuits

Gabriel Andrew Dirac
(1925-1984)

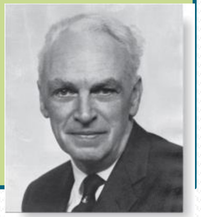
Unlike for an Euler circuit, no simple necessary and sufficient conditions are known for the existence of a Hamiton circuit.

However, there are some useful *sufficient 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. (*See the text for details*.)