8.3
Hamiltonian Paths and Circuits

8.3 – Hamiltonian Paths and Circuits

- A Hamiltonian path is a path that contains each vertex exactly once

- A Hamiltonian circuit is a Hamiltonian path that is also a circuit

8.3 – Hamiltonian Paths and Circuits

- Theorem 8.3.1
  - Let G be a connected graph with n vertices ($n \in Z$, $n > 2$), with no loops or multiple edges. G has a Hamiltonian circuit if for any 2 vertices u and v of G that are not adjacent, the degree of u plus the degree of v is $\geq n$.
- Corollary 8.3.1
  - G has a Hamiltonian circuit if each vertex has degree $\geq (n/2)$.

8.3 – Hamiltonian Paths and Circuits

- Theorem 8.3.2
  - Let the number of edges of G be m. Then G has a Hamiltonian circuit if $m \geq \frac{1}{2}(n^2 - 3n + 6)$ where n is the number of vertices.

Hamilton Paths and Circuits

A __Hamilton Path_____ is a continuous path that passes through every __vertex__ once and only once.

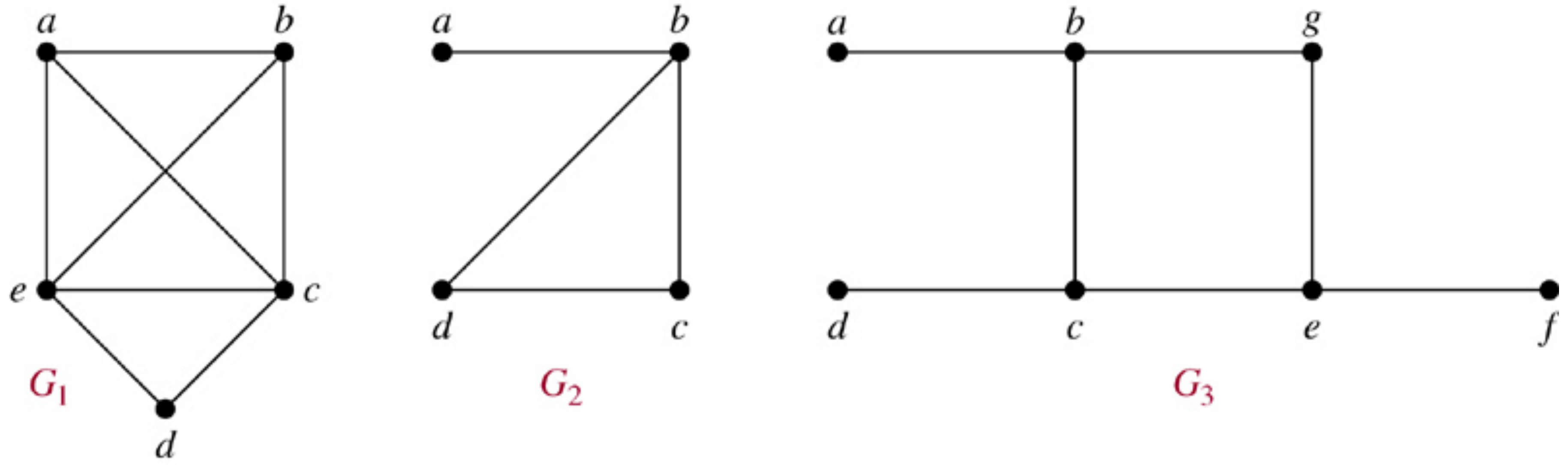
A Hamilton Circuit______ is a Hamilton path that begins and ends at the same vertex. (the starting/end vertex will be the only vertex touched twice

How is a Hamilton Path different from a Euler path or Circuit?

Hamilton circuits and paths

- A *Hamilton circuit* (*path*) is a simple circuit (path) that contains all vertices and passes through each vertex of the graph exactly once.
- How can we tell if a graph has a Hamilton circuit or path?
  - Not easily, i.e., in general, in not less than exponential time in the number of vertices
  - There are some tests that can exclude graphs (e.g., no vertex of degree one for circuits).

Ham. Circuits and Paths



Number of Hamilton paths

- Consider the complete graph K_n with $n > 2$. How many Hamilton circuits are there?
- Select any vertex as the start vertex (because all vertices will belong to the circuit the choice doesn't matter). From this vertex we can choose n-1 successor vertices, from each of them n-2 vertices, and so on, for a total of (n-1)! circuits. Because direction doesn't matter, the distinct circuits are (n-1)!/2.

Weighted graphs

- A *weighted* graph is a graph where numbers (*weights* or *costs*) have been attached to each edge.
- Example: In a graph for flight connections, weights could represent time needed for each flight, distance traveled, or cost of traveling on that flight. In a computer network, costs could represent the delay for a message to travel through a link.

Path length

- In a graph without weights, we define the length of a path as the number of edges in it.
- In a weighted graph, the path length is a function of the weights of the edges in the path, usually the sum of those weights.

The Traveling Salesman problem

- A traveling salesman needs to visit *n* cities, going to each city exactly once, and return to his starting city. Assume every city is connected to every other city, but the cost of traveling differs for each city pair. What is the sequence of cities that minimizes the overall cost?
- Equivalent formulation: Find the Hamilton circuit in K_n with shortest length.

Solving the TSP

- No algorithm is known that will guarantee finding the best solution except by examining all possible circuits (or equivalent approaches).
- For $n = 25$, $(n - 1)! / 2 = 24! / 2 \approx 3 \times 10^{23}$. If we take one nanosecond to calculate the cost of each circuit, we still need 3×10^{14} seconds, or about 10 million years.
- The problem is provably *NP-complete*.

TSP Problems

- Planning routes
  - 980 cities in Luxembourg
  - 24,978 cities in Sweden (solved May 2004)
  - 1,904,711 cities in the world
- Drilling printed circuit boards
  - cost of moving the drill and possibly changing drills because of different hole sizes
- Genome sequencing
  - edges represent likelihood that a fragment of DNA follows another fragment