Paths and Circuits (continued)

Recall:

An ***Eulerian trail*** in a pseudograph is a trail that passes through every vertex and includes every edge.

An ***Eulerian circuit*** in a pseudograph is a circuit that contains every vertex and every edge.

A pseudograph is ***Eulerian*** iff there exists an Eulerian circuit in it.

New definitions:

A ***Hamiltonian path*** in a graph is a path that passes through every vertex exactly once.

A ***Hamiltonian cycle*** in a graph is a cycle that contains every vertex of the graph. It is also called a ***Hamiltonian circuit***. (why?)

A ***Hamiltonian*** graph is a graph that has a Hamiltonian cycle.

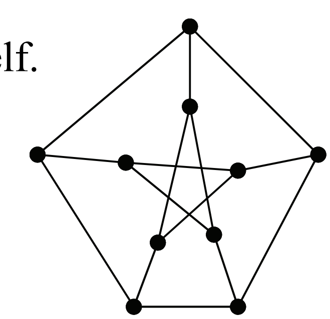
Properties:

If a vertex in a Hamiltonian graph has degree 2, then its both edges belong to a Hamiltonian cycle.

The only cycle contained in a cycle is the cycle itself.

Example

The ***Petersen graph*** is not Hamiltonian:



<u>Theorem (Dirac):</u>

If a graph has $n > 2$ vertices and every vertex has degree at least $n/2$, then the graph is Hamiltonian.

Application: ***gray codes***

<u>Definition:</u> the ***adjacency matrix*** of a graph of n vertices is the $n \times n$ binary matrix A with $A_{ij} = 1$ iff ij is an edge.

<u>Propositions:</u>

There is a one-to-one correspondence between ***labeled graphs*** (i.e., graphs with ordered sets of vertices) and symmetric binary matrices with zero diagonal.

Two graphs are isomorphic iff their vertices can be labeled in such a way that the corresponding adjacency matrices are equal.

Two labeled graphs are isomorphic iff there is a permutation of vertices that transforms one adjacency matrix into another.

An entry (i, j) of the matrix A^k is the number of walks of length k from i to j, $\forall\, k \in \mathbb{N}$.

<u>Definition:</u>

A ***weighted graph*** $G(V, E, w)$ is a graph $G(V, E)$ with a ***weight*** function (or ***distance***) $w: E \to [0, \infty)$.

The ***weight*** of a subgraph (e.g., a path) is the sum of weights of its edges.

Dijkstra's Algorithm (improved):

To find a shortest path in a weighted graph from vertex A to vertex E, do the following.

1. Assign to A the permanent label 0. Assign to every other vertex a temporary label ∞.
2. Repeat, until E is reached – or nothing changes:
    a. Take the vertex v that most recently acquired a permanent label d.
    b. For every vertex u adjacent to v that has no permanent label, change its temporary label t to $\min(t, d+w(u,v))$.
    c. Find a vertex in the graph with the smallest temporary label and make its label permanent.

Complexity = $O(n^2)$.

Floyd-Warshall Algorithm:

The task is: given the weight matrix W, compute the matrix of shortest path lengths.

For $k = 1$ to n,
    For $i = 1$ to n,
        For $j = 1$ to n,
            $W_{\mathrm{ij}} = \min(W_{ij}, W_{ik} + W_{kj})$

The output is W.

Complexity = $O(n^3)$.