Introduction to Graphs

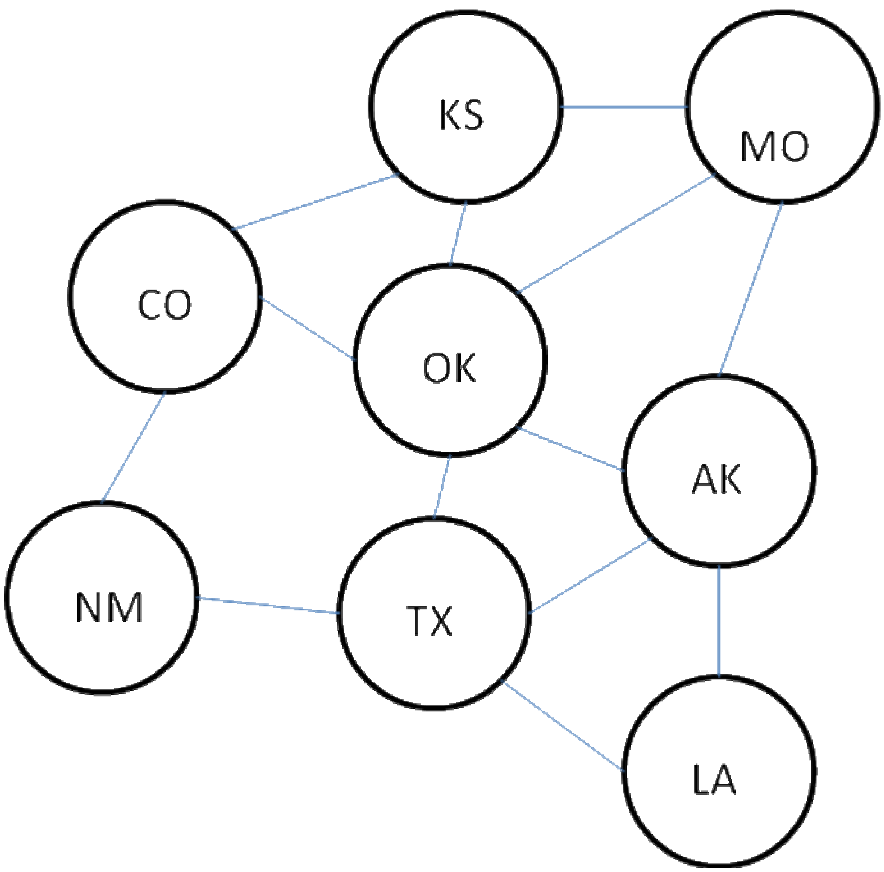
Graphs are an abstract data type. A graph consists of elements (normally referred to as vertices or nodes) connected by links (also known as edges).

Linked lists and binary trees are special cases of graphs.

Graphs are used throughout computer science to model things. Examples of real world things modeled using graphs include:

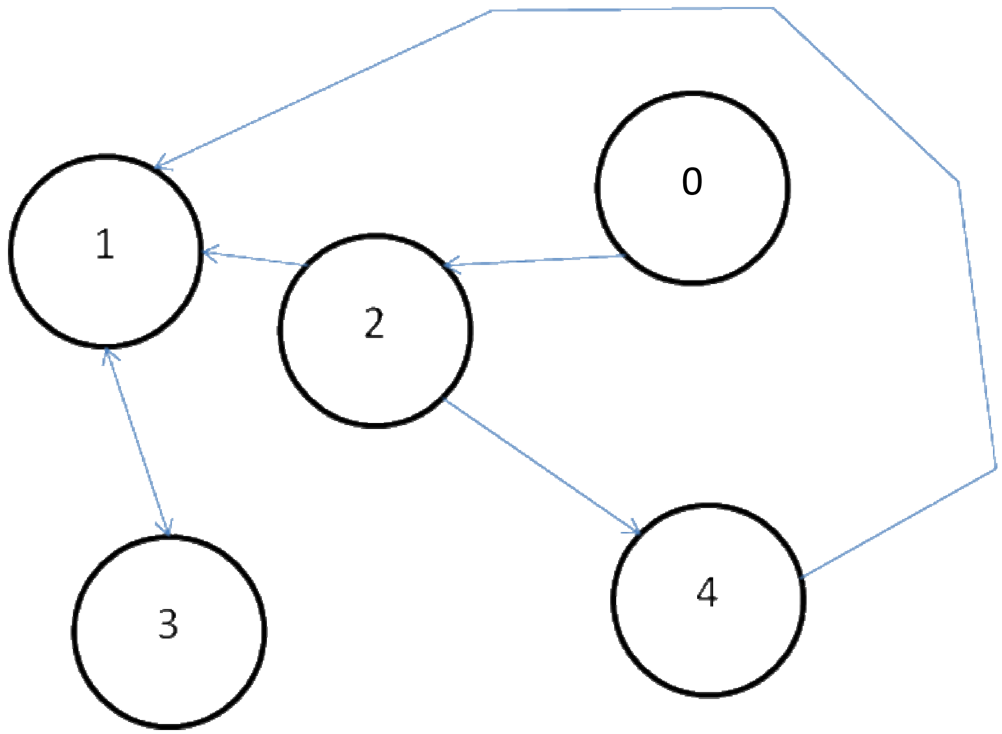
- computer networks
- cities connected by interstates
- airports connected by flights
- employees and emails sent between them

Here is an example graph that shows states as vertices. Edges exist between the states if they share a border.



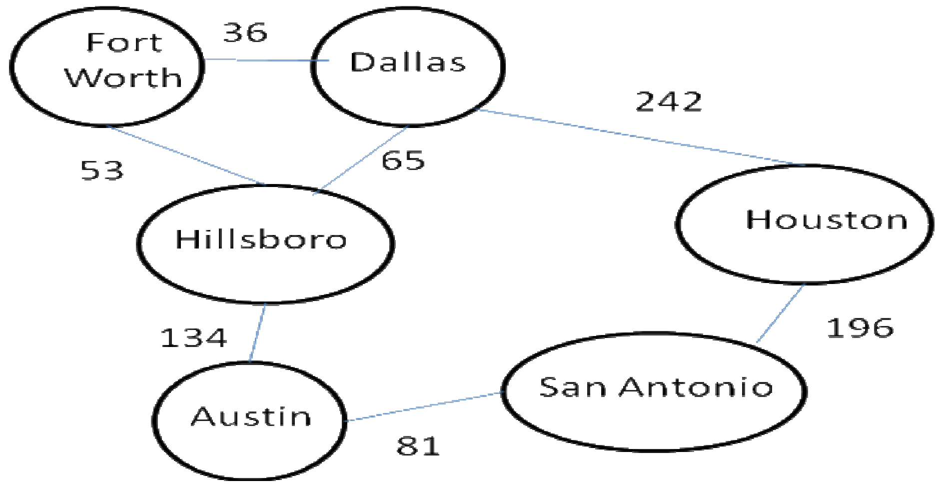
The graph shown above is undirected. This means movement is allowed along each direction of an edge between vertices.

Graphs may be directed. In this case, each edge will be labeled with the allowed direction of movement. Thus it may be possible to move from one vertex to another along an edge, but not back along the same edge. Here is an example of a directed graph.



The previous two graphs are unweighted graphs. In a weighted graph there is some number (usually a cost) associated with each edge.

For example, consider a graph with cities as vertices and edges as direct interstate highways between cities. The weight of each edge is the mileage between connected cities.



Interesting properties of graphs include the shortest path between two vertices and the diameter of the graph. In the graph shown above the shortest path between Austin and Houston has a weight of 277. (Austin to San Antonio to Houston)

The diameter of a graph is the longest of the shortest paths that exist between all the vertices in a graph.

The two common ways of representing graphs are via an adjacency list or an adjacency matrix.

In the adjacency list approach, each node is stored in a list. Each node contains its own list of the nodes it connects to.

In the adjacency matrix approach a matrix with N nodes with an N by N matrix of booleans. If a connection exists from node i to node j then element i,j in the matrix is true, otherwise it is false. The following adjacency matrix represents the directed graph at the top of page 2.

false	false	true	false	false
false	false	false	true	false
false	true	false	false	true
false	true	false	false	false
false	true	false	false	false

References

Black, P. E. (1998, September 4). *Dictionary of Algorithms and Data Structures*. Retrieved August 14, 2010, from National Institute of Standards and Technology: http://www.itl.nist.gov/div897/sqg/dads/