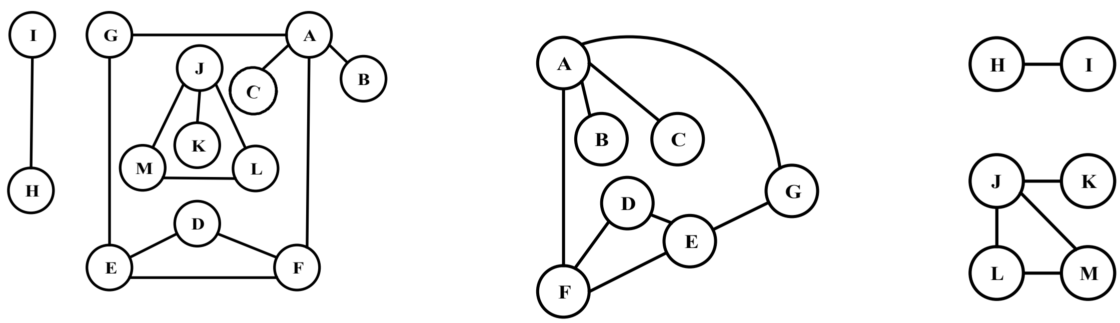
Graph Theory

This description is adapted from Robert Sedgewick's *Algorithms*, (Addison-Wesley, 1983). This book, a reference for virtually all genres of computer algorithms, contains many example programs in Pascal. Moreover, it is lucid and well written. We recommend this book highly.

Many problems are naturally formulated in terms of points and connections between them. For example, an electric circuit has gates connected by wires, an airline map has cities connected by routes, and a program flowchart has boxes connected by arrows. A graph is a mathematical object which models such situations.

A *graph* is a collection of vertices and edges. An *edge* is a connection between two *vertices* (or *nodes*). One can draw a graph by marking points for the vertices and drawing lines connecting them for the edges, but it must be borne in mind that the graph is defined independently of the representation. For example, the following two drawings represent the same graph: (page 374)



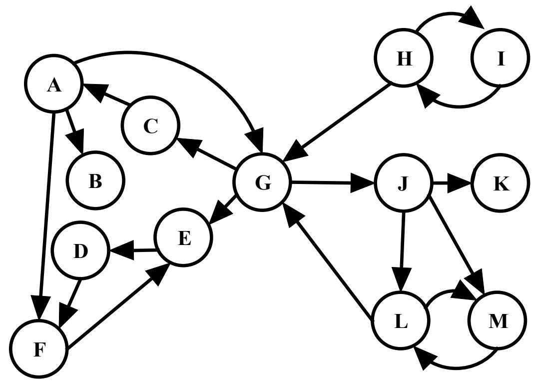
The precise way to represent this graph is to say that it consists of the set of vertices {*A, B, C, D, E, F, G, H, I, J, K, L, M*}, and the set of edges between these vertices {*AG, AB, AC, LM, JM, JL, JK, ED, FD, HI, FE, AF, GE*}.

A *path* from vertex *x* to *y* in a graph is a list of vertices, in which successive vertices are connected by edges in the graph. For example, *BAFEG* is path from *B* to *G* in the graph above. A *simple path* is a path with no vertex repeated. For example, *BAFEGAC* is not a simple path.

A graph is *connected* if there is a path from every vertex to every other vertex in the graph. Intuitively, if the vertices were physical objects and the edges were strings connecting them, a connected graph would stay in one piece if picked up by any vertex. A graph which is not connected is made up of *connected components*. For example, the graph above has three connected components: {*I, H*}, {*J, K, L, M*} and {*A, B, C, D, E, F, G*}.

A *cycle* is a path, which is simple except that the first and last vertex are the same (a path from a point back to itself). For example, the path *AFEGA* is a cycle in our example. Vertices must be listed in the order that they are traveled to make the path; any of the vertices may be listed first. Thus, *FEGAF* and *GAFEG* are different ways to identify the same cycle. For clarity, we list the start / end vertex twice: once at the start of the cycle and once at the end. A graph with no cycles is called a *tree*. There is only one path between any two nodes in a tree. A tree on N vertices contains exactly N-1 edges. A *spanning tree* of a graph is a subgraph that contains all the vertices and forms a tree. A group of disconnected trees is called a *forest*.

Directed graphs are graphs which have a direction associated with each edge. An edge *xy* in a directed graph can be used in a path that goes from *x* to *y* but not necessarily from *y* to *x*. For example, a directed graph similar to our example graph is drawn below. (page 422)



There is only one directed path from D to F. Note that there are two edges between H and I, one each direction, which essentially makes an undirected edge. An undirected graph can be thought of as a directed graph with all edges occurring in pairs in this way. A *dag* (*directed acyclic graph*) is a directed graph with no cycles.

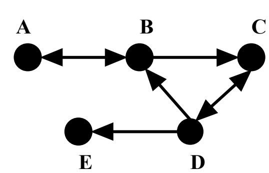
We'll denote the number of vertices in a given graph by V, the number of edges by E. Note that E can range anywhere from V to V^2 (or 1/2 $V(V-1)$ in an undirected graph). Graphs will all edges present are called *complete* graphs; graphs with relatively few edges present (say less than $V \log(V)$) are called *sparse*; graphs with relatively few edges missing are called *dense*.

It is frequently convenient to represent a graph by a matrix, as shown in the second sample problem below. If we consider vertex A as 1, B as 2, etc., then a "one" in M at row i and column j indicates that there is a path from vertex i to j. If we raise M to the pth power, the resulting matrix indicates which paths of length p exist in the graph. In fact, the quantity $M^p(i,j)$ is the number of paths.

References

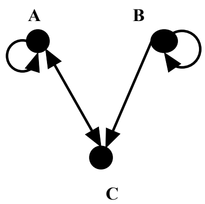
Ore, Oystein. *Graphs and Their Uses*, MAA New Mathematic Library #10 (1963).
Sedgewick, Robert. *Algorithms*. Addison-Wesley (1983).

Sample Problems

<table>
<tr>
<td>Find the number of different cycles contained in the directed graph with vertices
{A,B,C,D,E}
and edges
{AB,BA,BC,CD,DC,DB,DE}.</td>
<td>The graph is as follows:


By inspection, the cycles are: {A,B}, {B,C,D} and {C,D}. Thus, there are 3 cycles in the graph.</td>
</tr>
</table>

In the following directed graph, find the total number of different paths from vertex A to vertex C of length 2 or 4.



By inspection, the only path of length 2 is A→A→C. The paths of length 4 are: A→A→A→A→C, A→A→C→A→C and A→C→A→A→C.

Alternatively, let matrix M represent the graph. Recall that the number of paths from vertex *i* to vertex *j* of length p equals $M^p(i,j)$. The values of M, M^2 and M^4 are:

$$\begin{vmatrix}101\\011\\100\end{vmatrix}, \begin{vmatrix}201\\111\\101\end{vmatrix}, \begin{vmatrix}503\\413\\302\end{vmatrix}$$

There is 1 path of length 2 ($M^2(1,3)$) and 3 paths of length 4 ($M^4(1,3)$).