Graph Theory

Victor Adamchik

Fall of 2005

■ Plan

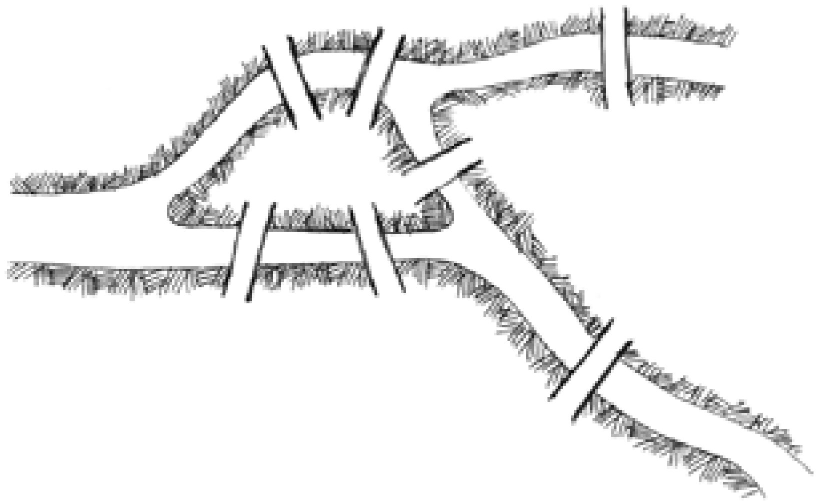
1. Basic Vocabulary
2. Regular graph
3. Connectivity
4. Representing Graphs

■ Introduction

A.Aho and J.Ulman acknowledge that “Fundamentally, computer science is a science of abstraction.” Computer scientists must create abstractions of real-world problems that can be represented and manipulated in a computer.

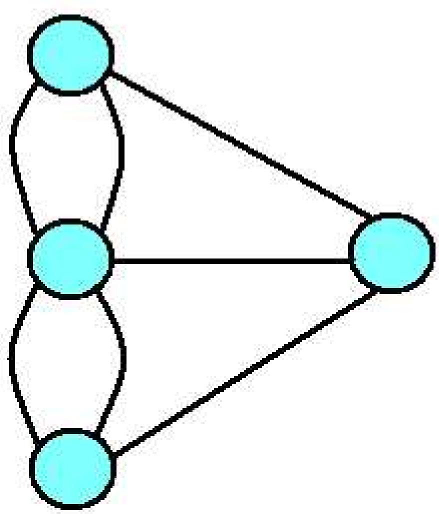
Sometimes the process of abstraction is simple. For example, we use a logic to design a computer circuits. Another example - scheduling final exams. For successful scheduling we have to take into account associations between courses, students and rooms. Such set of connections between items is modeled by *graphs*. Let me reiterate, in our model the set of items (courses, students and rooms) won't be much helpful. We also have to have a set of connections between pairs of items, because we need to study the relationships between connections.

The basic idea of graphs were introduced in 18th century by the great Swiss mathematician Leonhard Euler. He used graphs to solve the famous Königsberg bridge problem. Here is a picture (taken from the internet)



German city of Königsberg (now it is Russian Kaliningrad) was situated on the river Pregel. It had a park situated on the banks of the river and two islands. Mainland and islands were joined by seven bridges. A problem was whether it was possible to take a walk through the town in such a way as to cross over every bridge once, and only once.

Here is the graph model of the problem



A graph is a set of points (we call them vertices or nodes) connected by lines (edges or arcs). The simplest example known to you is a linked list.

The Web: The entire Web is a graph, where items are documents and the references (links) are connections.

Networks: A network consist of sites that send and recieve messages of various types.

Program Structure: A compiler builds a graph to represent relationships between classes. The items are classes; connections are represented by possibility of a method of one class to call a method from another class

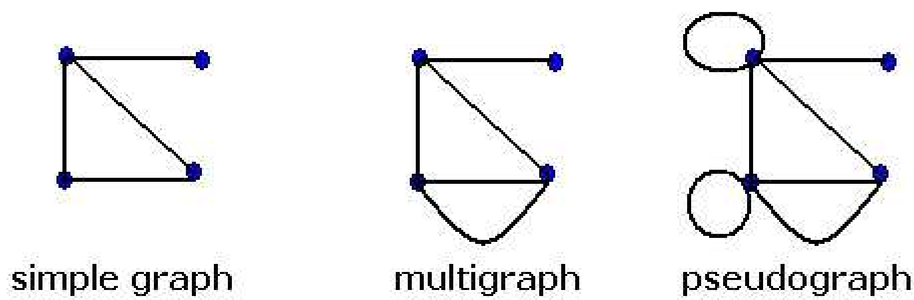
Basic Vocabulary

A substantial amount of definitions is associated with graphs. We start with introduction to different types of graphs

A graph that have nonempty set of vertices connected at most by one edge is called simple

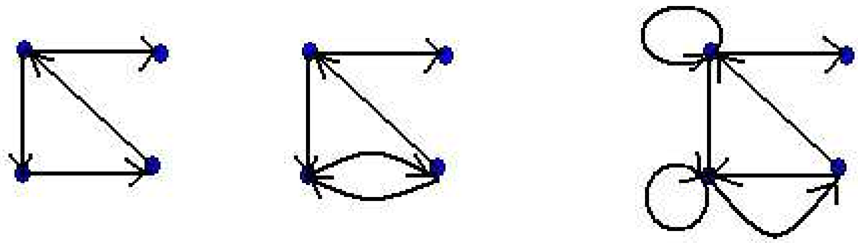
When simple graphs are not efficient to model a cituation, we consider multigraphs. They allow multiple edges between two vertices.

If that is not enough, we consider pseudographs. They allow edges connect a vertex to itself.



What do these three types of graphs have in common? The set of edges is *unordered*. All such graphs are called undirected.

A directed graph consist of vertices and ordered pairs of edges. Note, multiple edges in the same direction are not allowed.

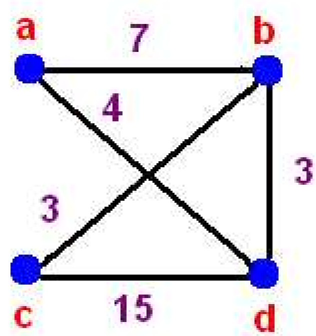


If multiple edges in the same direction are allowed, then a graph is called directed multigraph.

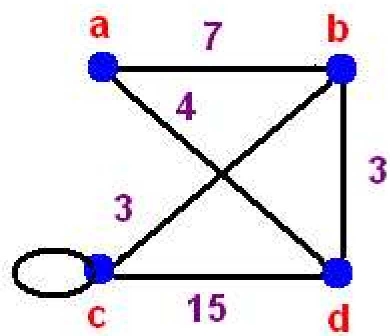
Usually by a graph people mean a simple undirected graph. No directions, no self-loops, no multiple edges. Be careful and watch out!

An edge may also have a weight or cost associated with it.

If (a, b) is an edge we might denote the cost by $c(a, b)$ In the example below, $c(a, b) = c(b, a) = 7$.



Two vertices are called adjacent if there is an edge between them. The degree of a vertex in an undirected graph is the number of edges associated with it. If a vertex has a loop, it contributes twice.



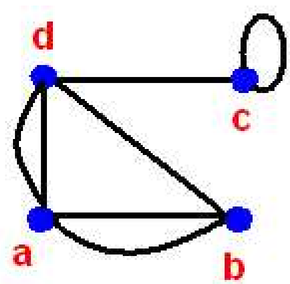
In the above picture, the degree of vertex a is 2, and the degree of vertex c is 4.

Theorem (The handshaking theorem).

Let G be an undirected graph (or multigraph) with V vertices and N edges. Then

$$2N = \sum_{v=V} deg(v)$$

Example,

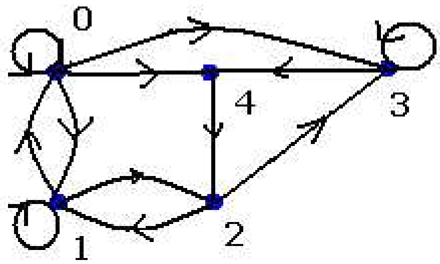


Exercise.

Suppose a simple graph has 15 edges, 3 vertices of degree 4, and all others of degree 3. How many vertices does the graph have? *3*4 + (x-3)*3 = 30*

In a directed graph terminology reflects the fact that each edge has a direction.

In a directed graph vertex v is adjacent to u, if there is an edge leaving v and coming to u. In a directed graph the in-degree of a vertex denotes the number of edges coming to this vertex. The out-degree of a vertex is the number of edges leaving the vertex.



in-deg(0) = 2, in-deg(1) = 3, in-deg(2) = 2, in-deg(3) = 3, in-deg(4) = 2

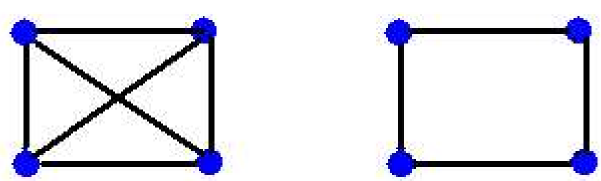
Theorem.

Let G be a directed graph (or multigraph) with V vertices and N edges. Then

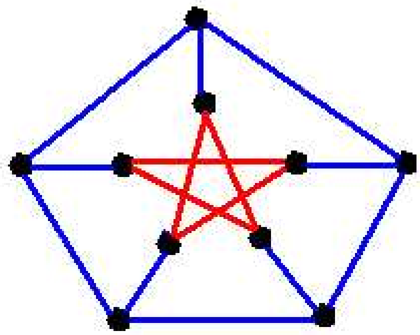
$$N = \sum_{v=V} indeg(v) = \sum_{v=V} outdeg(v)$$

Regular graph

A graph in which every vertex has the same degree is called a ***regular*** graph. Here is an example of two regular graphs with four vertices that are of degree 2 and 3 correspondently



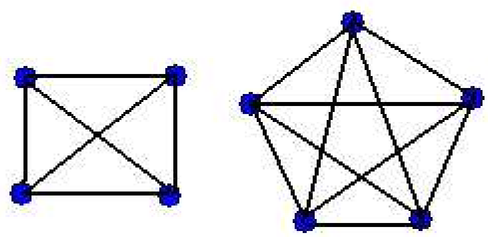
The following graph of degree 3 with 10 vertices is called the Petersen graph (after Julius Petersen (1839-1910), a Danish mathematician.)



Exercise.
Given a regular graph of degree d with V vertices, how many edges does it have?

The complete graph on n vertices, denoted K_n, is a simple graph in which there is an edge between every pair of distinct vertices.

Here are K_4 and K_5:



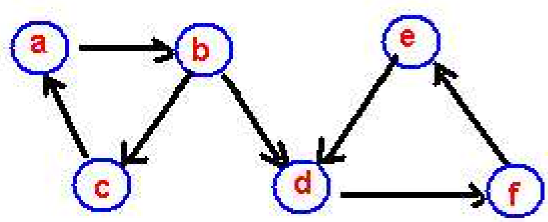
Exercise.How many edges in K_n?

Connectivity

A path is a sequence of distinctive vertices connected by edges. Vertex v is reachable from u if there is a path from u to v. A graph is connected, if there is a path between any two vertices.

Exercise.
Given a graph with 7 vertices; 3 of them of degree two and 4 of degree one. Is this graph is connected?
No, the graph have 5 edges.

A directed graph is strongly connected if there is a path from u to v and from v to u for any u and v in the graph. A directed graph is weakly connected if the underlying undirected graph is connected



Representing Graphs

Theorem.

In an undirected simple graph with N vertices, there are at most $\frac{N(N-1)}{2}$ *edges.*

Proof. By induction on the number of vertices.

V = 1, there are no edges

V = n, there are $n(n-1)/2$ edges

We need to prove that if $V = n+1$ then a graph has $n(n+1)/2$ edges

$$\frac{n(n-1)}{2} + n = \frac{n(n+1)}{2}$$

Exercise.

What is the maximum number of edges in a simple disconnected graph with N vertices?

For all graphs, the number of edges E and vertices V satisfies the inequality $E < V^2$.

If the number of edges is close to $V * \log V$, we say that this is a dense graph, it has a large number of edges. Otherwise, this is a sparse graph $E < V * \log V$. In most cases, the graph is relatively sparse.

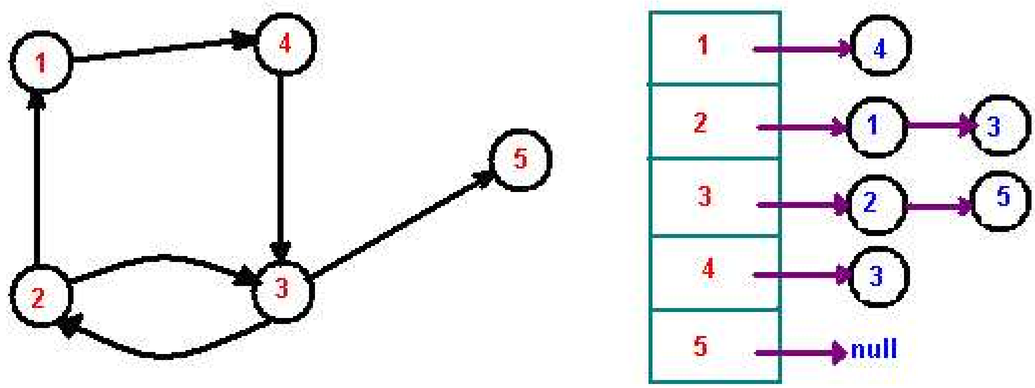
There are two standard ways to represent a graph:

- as a collection of adjacency lists

- or as an adjacency matrix

An adjacency list representation is used for representation of the sparse graphs. An adjacency matrix representation may be preferred when the graph is dense.

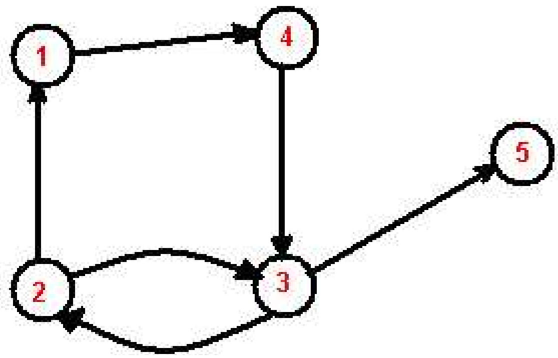
The adjacency-list representation of a graph G consists of an array of linked lists, one for each vertex. Each such list contains all vertices adjacent to a chosen one. Here is an adjacency-list representation:



Vertices in an adjacency list are stored in an arbitrary order. A potential disadvantage of the adjacency-list representation is that there is no quicker way to determine if there is an edge between two given vertices.

This disadvantage is eliminated by an adjacency matrix representation. The adjacency matrix is a matrix of size $V \; x \; V$ such that

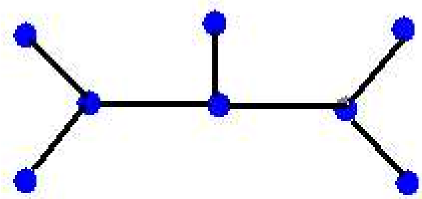
$$M_{i\,j} = \begin{cases} 1, & \text{if there is an edge between } i \text{ and } j \\ 0, & \text{otherwise} \end{cases}$$



0	0	0	1	0
1	0	1	0	0
0	1	0	0	1
0	0	1	0	0
0	0	0	0	0

Trees

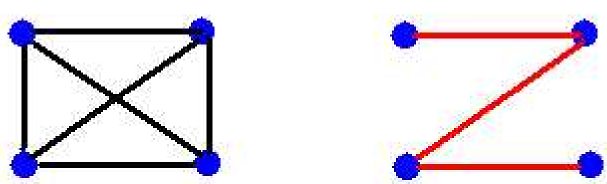
A tree is a connected simple graph without cycles. A cycle is a sequence of distinctive adjacent vertices that begins and ends at the same vertex.



A tree with V vertices must have $V-1$ edges. A connected graph with V vertices and $V-1$ edges must be a tree.

A rooted tree is a tree with one vertex designated as a root. A forest is a graph without cycles. In orther words, a forest is a set of trees.

A spanning tree of a graph is a subgraph, which is a tree and contains all vertices of the graph. In the figure below, the right picture represents a spanning tree for the graph on the left. A spanning tree is not unique.



Famous Problems on Graphs

The Euler cycle (or tour) problem: Is it possible to traverse each of the edges of a graph exactly once, starting and ending at the same vertex?

The Hamiltonian cycle problem: Is it possible to traverse each of the vertices of a graph exactly once, starting and ending at the same vertex?

The traveller salesman problem: Find the shortest path in a graph that visits each vertex at least once, starting and ending at the same vertex?

The planar graph: Is it possible to draw the edges of a graph in such a way that edges do not cross?

The four coloring problem: Is it possible to color the vertices of a graph with at most 4 colors such that adjacent vertices get different color?

The marriage problem (or bipartite perfect matching): At what condition a set of boys will be marrying off to a set of girls such that each boy gets a girl he likes?