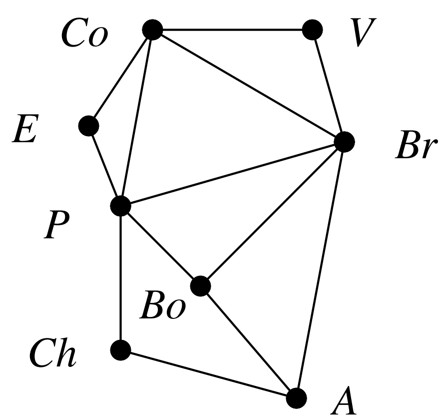
Section 1.4 Graphs and Trees

A *graph* is set of objects called *vertices* or *nodes* where some pairs of objects may be connected by *edges*. (A *directed graph* has edges that point in one direction.)

Example. Draw a graph of the South American countries that touch the Pacific Ocean and their neighbors, where the vertices are countries and an edge indicates a common border.



Vertices = {*Co, V, E, Br, P, Bo, Ch, A*}

Edges = {{*Co, V*}, {*Co, E*}, ….}.

A *path* from vertex x_0 to x_n is a sequence of edges that we denote by vertices $x_0, x_1, \ldots, x_n$, where there is an edge from x_{i-1} to x_i for $1 \leq i \leq n$.

The *length* of a path is the number of edges.

A *cycle* is a path with distinct edges that begins and ends at the same vertex.

Example. *A, Bo, A,* is not a cycle since the edge {*A, Bo*} occurs twice. *A, Bo, Br, A,* is a cycle.

Quiz (1 minute). What is a longest path from *A* to *V* with distinct edges and no cycles?

Answer: The length is 6. For example, *A, Bo, Br, P, E, Co, V*.

A graph is *n-colorable* if it's vertices can be colored with *n* colors with distinct colors for adjacent vertices. The *chromatic number* of a graph is the smallest such *n*.

Quiz (1 minute). What is the chromatic number of the example graph?

Graph Traversals

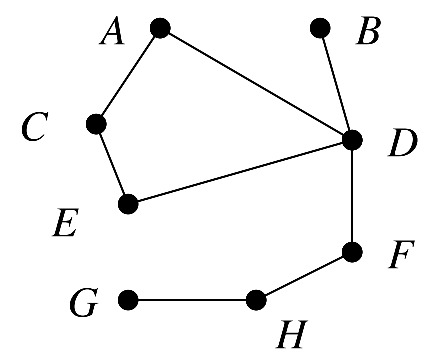
A graph traversal starts at some vertex v and visits all vertices x that can be reached by a path from v to x. But don't visit any vertex more than once.

Breadth-First: If the graph has n vertices then start with a vertex v and do the following:

for $k := 0$ **to** $n - 1$ **do** visit(v, k) **od**

where visit(v, k) visits all x not visited if there is a path from v to x of length k.

Use the pictured graph for the following quizzes.



Quiz (1 minute). Find a breadth-first traversal that starts at *F*.

One answer: *F, H, D, G, B, A, E, C*.

Quiz (1 minute). Find a breadth-first traversal that starts at *C*.

One answer: *C, A, E, D, B, F, H, G*.

Depth-First: Start at a vertex v and call the procedure $D(v)$, which is defined as follows:

$D(v)$: **if** v has not been visited **then**
visit(v);
for each edge from v to x **do** $D(x)$ **od**
fi

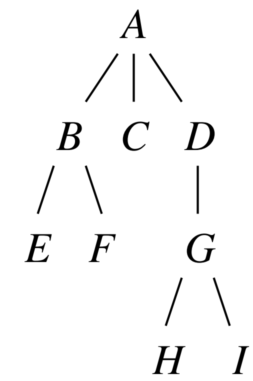
Quiz (1 minute). Find a depth-first traversal of the pictured graph that starts at *F*.

One answer: *F, H, G, D, B, A, C, E*.

Quiz (1 minute). Find a depth-first traversal of the pictured graph that starts at *E*.

One answer: *E, D, F, H, G, A, C, B*.

Trees



A *tree* is a connected graph (a path between any two points) with no cycles. Most trees are oriented so that they look like upside-down trees, such as the tree pictured.

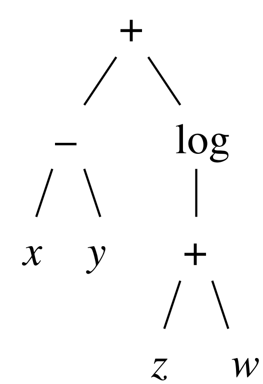
The top node is the *root*, the nodes directly below a node are its *children*, the node directly above a node is the *parent*, the bottom nodes are *leaves*, and the *height* or *depth* of the tree is the length of the longest path of distinct edges from root to a *leaf*.

Example. For this tree the root is A. The children of A are B, C, D. D is the parent of G. The height or depth of the tree is 3. The leaves are E, F, C, H, I.

Any node of a tree is the root of a *subtree*. One way to represent a tree is as a list whose head is the root of the tree and whose tail is the list of subtrees, where each subtree is represented in the same way.

Example. The pictured tree can be represented by the list

$$\langle A, \langle B, \langle E\rangle, \langle F\rangle\rangle, \langle C\rangle, \langle D, \langle G, \langle H\rangle, \langle I\rangle\rangle\rangle\rangle.$$



Any algebraic expression can be represented as a tree. For example, the tree for the expression $(x - y) + \log(z + w)$ is pictured to the right.

Quiz (1 minute). Do a depth-first (left to right) traversal.

Answer: $+ - x\ y \log + z\ w$. This is the prefix form of the expression.

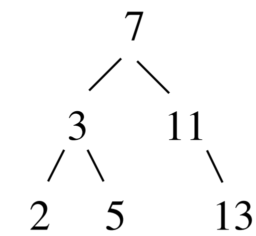
Binary Trees

A *binary tree* is either *empty*, denoted by ⟨⟩, or each node has two subtrees that are binary trees and are called the *left* and *right* subtrees of the node. If a binary tree is not empty, we'll represent it as a list of the form ⟨*L*, *x*, *R*⟩, where *x* is the root and *L* and *R* are the left and right subtrees, respectively.

Example. The binary tree with a single node *x* is denoted by ⟨⟨⟩, *x*, ⟨⟩⟩.

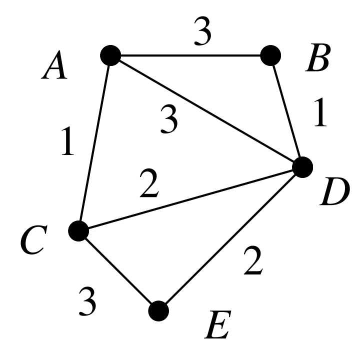
A *binary search tree* represents ordered information, where the predecessors and successors of a node are in its left and right subtrees, respectively.

Example. A binary search tree for the first six prime numbers is pictured.



Spanning Trees

A *spanning tree* for a connected graph is a tree whose nodes are the nodes of the graph and whose edges are a subset of the edges of the graph. A *minimal spanning tree* minimizes the sum of weights on the edges of all spanning trees.



Example. Use Prim's algorithm to construct a minimal spanning tree for the pictured graph, starting with node *D*.

Solution: A minimal spanning tree is constructed in 4 steps: