Trees in Graphs

Graphs

- Graph consists of two sets: set V of vertices and set E of edges.
- Terminology: endpoints of the edge, loop edges, parallel edges, adjacent vertices, isolated vertex, subgraph, bridge edge
- Directed graph (digraph) has each edge as an ordered pair of vertices

Special Graphs

- Simple graph is a graph without loop or parallel edges. A complete graph of n vertices K_n is a simple graph which has an edge between each pair of vertices. A complete bipartite graph of (n, m) vertices $K_{n,m}$ is a simple graph consisting of vertices, v_1, v_2, …, v_m and w_1, w_2, …, w_n with the following properties:
  - There is an edge from each vertex v_i to each vertex w_j
  - There is no edge from any vertex v_i to any vertex v_j
  - There is no edge from any vertex w_i to any vertex w_j

The Concept of Degree

- The degree of a vertex deg(v) is a number of edges that have vertex v as an endpoint. Loop edge gives vertex a degree of 2. In any graph the sum of degrees of all vertices equals twice the number of edges. The total degree of a graph is even. In any graph there are even number of vertices of odd degree

Paths and Circuits

- A walk in a graph is an alternating sequence of adjacent vertices and edges. A path is a walk that does not contain a repeated edge. Simple path is a path that does not contain a repeated vertex. A closed walk is a walk that starts and ends at the same vertex. A circuit is a closed walk that does not contain a repeated edge. A simple circuit is a circuit which does not have a repeated vertex except for the first and last

Connectedness

- Two vertices of a graph are connected when there is a walk between two of them. The graph is called connected when any pair of its vertices is connected. If graph is connected, then any two vertices can be connected by a simple path. If two vertices are part of a circuit and one edge is removed from the circuit then there still exists a path between these two vertices. Graph H is called a connected component of graph G when H is a subgraph of G, H is connected and H is not a subgraph of any bigger connected graph. Any graph is a union of connected components

Euler Circuit

- Euler circuit is a circuit that contains every vertex and every edge of a graph. Every edge is traversed exactly once. If a graph has Euler circuit then every vertex has even degree. If some vertex of a graph has odd degree then the graph does not have an Euler circuit. If every vertex of a graph has even degree and the graph is connected then the graph has an Euler circuit. A Euler path is a path between two vertices that contains all vertices and traverces all edge exactly ones. There is an Euler path between two vertices v and w iff vertices v and w have odd degrees and all other vertices have even degrees

Hamiltonian Circuit

Hamiltonian circuit is a simple circuit that contains all vertices of the graph (and each exactly once). Example: Traveling salesperson problem

Trees

- Connected graph without circuits is called a tree. Graph is called a forest when it does not have circuits. A vertex of degree 1 is called a terminal vertex or a leaf, the other vertices are called internal nodes. Examples: Decision tree, Syntactic derivation tree.
- Any tree with more than one vertex has at least one vertex of degree 1. Any tree with n vertices has n – 1 edges. If a connected graph with n vertices has n – 1 edges, then it is a tree

Rooted Trees

- Rooted tree is a tree in which one vertex is distinguished and called a root. Level of a vertex is the number of edges between the vertex and the root. The height of a rooted tree is the maximum level of any vertex. Children, siblings and parent vertices in a rooted tree. Ancestor, descendant relationship between vertices

Binary Trees

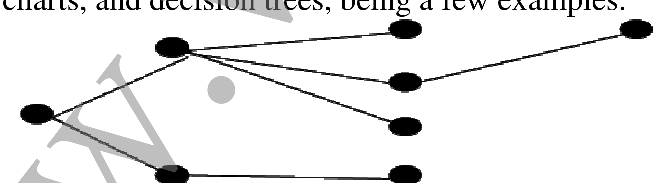
- Binary tree is a rooted tree where each internal vertex has at most two children: left and right. Left and right subtrees.
- Full binary tree: Representation of algebraic expressions
- If T is a full binary tree with k internal vertices then T has a total of 2k + 1 vertices and k + 1 of them are leaves. Any binary tree with t leaves and height h satisfies the following inequality: $t \leq 2^h$

Spanning Trees

- A subgraph T of a graph G is called a spanning tree when T is a tree and contains all vertices of G. Every connected graph has a spanning tree. Any two spanning trees have the same number of edges. A weighted graph is a graph in which each edge has an associated real number weight. A minimal spanning tree (MST) is a spanning tree with the least total weight of its edges.

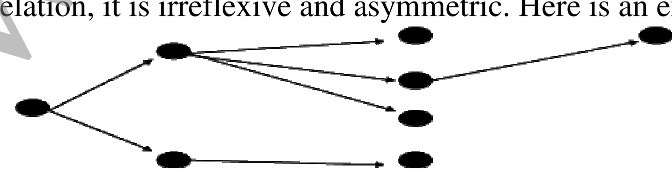
Trees: Definition & Applications

A *tree* is a connected graph with no cycles. A *forest* is a graph whose components are trees. An example appears below. Trees come up in many contexts: tournament brackets, family trees, organizational charts, and decision trees, being a few examples.



Directed Trees

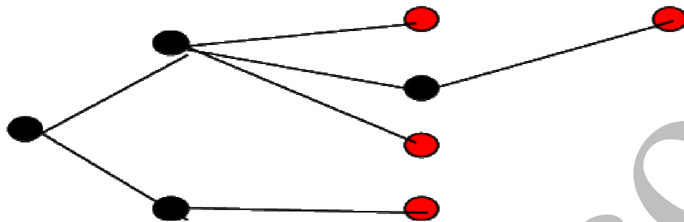
A *directed tree* is a digraph whose underlying graph is a tree and which has no loops and no pairs of vertices joined in both directions. These last two conditions mean that if we interpret a directed tree as a relation, it is irreflexive and asymmetric. Here is an example.



Theorem: A tree T(V,E) with finite vertex set and at least one edge has at least two leaves (a *leaf* is a vertex with degree one). Proof: Fix a vertex a that is the endpoint of some edge. Move from a to the adjacent vertex along the edge. If that vertex has no adjacent vertices then it has degree one, so stop. If not, move along another edge to another vertex. Continue building a path in this fashion until you reach a vertex with no adjacent vertices besides the one you just came from. This is sure to happen because V is finite and you never use the same vertex twice in the path (since T is a tree). This produces one leaf. Now return to a. If it is a leaf, then you are done. If not, move along a different edge than the one at the first step above. Continue extending the path in that direction until you reach a leaf (which is sure to happen by the argument above).

Trees: Leaves & Internal Vertices

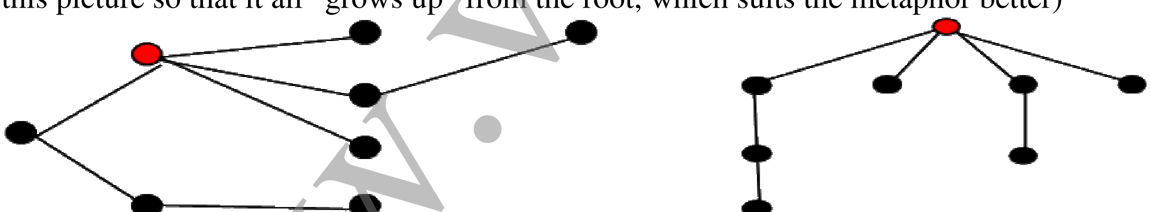
In the following tree the red vertices are leaves. We now know every finite tree with an edge has a least two leaves. The other vertices are *internal vertices*.



- Theorem: Given vertices a and b in a tree T(V,E), there is a unique simple path from a to b. Proof: Trees are connected, so there is a simple path from a to b. The book gives a nice example of using the contrapositive to prove the rest of the theorem.
- Theorem: Given a graph G(V,E) such that every pair of vertices is joined by a unique simple path, then G is a tree. This is the converse of Theorem 6.37. Proof: Since a simple path joins every pair of points, the graph is connected. Now suppose G has a cycle abc…a. Then ba and bc…a are distinct simple paths from b to a. This contradicts uniqueness of simple paths, so G cannot possess such a cycle. This makes G a tree.

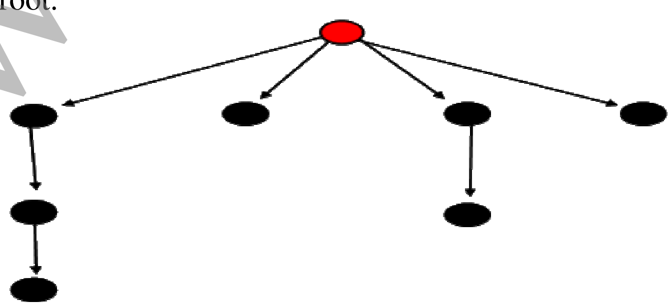
Rooted Trees

Sometimes it is useful to distinguish one vertex of a tree and call it the *root* of the tree. For instance we might, for whatever reasons, take the tree above and declare the red vertex to be its root. In that case we often redraw the tree to let it all "hang down" from the root (or invert this picture so that it all "grows up" from the root, which suits the metaphor better)



Rooted Directed Trees

It is sometimes useful to turn a rooted tree into a rooted directed tree T′ by directing every edge away from the root.



Rooted trees and their derived rooted directed trees have some useful terminology, much of which is suggested by family trees. The *level* of a vertex is the length of the path from it to the root. The *height* of the tree is the length of the longest path from a leaf to the root. If there is a directed edge in T′ from a to b, then a is the *parent* of b and b is a *child* of a. If there are directed edges in T′ from a to b and c, then b and c are *siblings*. If there is a directed path from a to b, then a is an *ancestor* of b and b is a *descendant* of a.

Binary & m-ary Trees

We describe a directed tree as *binary* if no vertex has outdegree over 2. It is more common to call a tree *binary* if no vertex has degree over 3. (In general a tree is m-ary if no vertex has degree over m+1. Our book calls a directed tree m-ary if no vertex has outdegree over m.) The directed rooted tree above is 4-ary (I think the word is quaternary) since it has a vertex with outdegree 4. In a rooted binary tree (hanging down or growing up) one can describe each child vertex as the *left child* or *right child* of its parent.

Trees: Edges in a Tree

Theorem: A tree on n vertices has n–1 edges. Proof: Let T be a tree with n vertices. Make it rooted. Then every edge establishes a parent-child relationship between two vertices. Every child has exactly one parent, and every vertex except the root is a child. Therefore there is exactly one edge for each vertex but one. This means there are n–1 edges.

Theorem: If G(V,E) is a connected graph with n vertices and n–1 edges is a tree.
Proof: Suppose G is as in the statement of the theorem, and suppose G has a cycle. Then we can remove an edge from the cycle without disconnecting G (see the next slide for why). If this makes G a tree, then stop. If not, there is still a cycle, so we can remove another edge without disconnecting G. Continue the process until the remaining graph is a tree. It still has n vertices, so it has n–1 edges by a prior theorem. This is a contradiction since G had n–1 vertices to start with. Therefore G has no cycle and is thus a tree.

(Why can we remove an edge from a cycle without disconnecting the graph? Let a and b be vertices. There is a simple path from a to b. If the path involves no edges in the cycle, then the path from a to be is unchanged. If it involves edges in the cycle, let x and y be the first and last vertices in the cycle that are part of the path from a to b. So there is a path from a to x and a path from y to b. Since x and y are part of a cycle, there are at least simple two paths from x to y. If we remove an edge from the cycle, at least one of the paths still remains. Thus there is still a simple path from a to b.)

Spanning Trees of a Graph

If G(V,E) is a graph and T(V,F) is a subgraph of G and is a tree, then T is a *spanning tree* of G. That is, T is a tree that includes every vertex of G and has only edges to be found in G. Using the procedure in the previous paragraph (remove edges from cycles until only a tree remains), we can easily prove that every connected graph has a spanning tree.