EE360C	Spring 2007

Basic Definitions

Adnan Aziz — *UT Austin*

Appendix B
You should be comfortable with the definitions related to sets (CLRS B.1), relations (CLRS B.2), and functions (CLRS B.3).

Graphs

CLRS Chapter B.4
Two kinds of graphs—directed and undirected

- minor differences across textbooks
- will postpone the problem of how to best represent graphs in memory

A *directed graph* (also called a *digraph*) G is a pair (V, E) where V is a finite set (of "vertices") and E (of "edges") is a subset of $V \times V$.

- allows for self loops

An *undirected graph* G is a pair (V, E) where V is a finite set (of "vertices") and E (of "edges") is a set of unordered pairs of edges $\{u, v\}$, where $u \neq v$.

- observe—no self loops
- common to still write (u, v)

Many definitions are very similar for directed and undirected graphs.
Notation—I will *emphasize* defined terms.
If (u, v) is an edge in a digraph G, we'll say (u, v) is *incident from* or *leaves* u and is *incident to* or *enters* v.
If (u, v) is an edge in an undirected graph G, we'll say (u, v) is *incident to* the vertices u and v.

In either case, we say v is *adjacent* to vertex u; in the case of a digraph this relation is not necessarily symmetric.
The *degree* of a vertex in an undirected graph is the number of edges incident to it.
The *out-degree* of a vertex in a digraph is the number of edges leaving it; the *in-degree* of a vertex in a digraph is the number of edges entering it.
A *path* from a vertex u to a vertex v is a sequence of vertices $\langle v_0, v_1, \ldots, v_k \rangle$ such that $u = v_0$, $u = v_k$, and $v_{i-1}, v_i) \in E$ for $i = 1, 2, \ldots, k$.

- *length* of a path = number of edges
- the path above *contains* the vertices $v_0, v_1, \ldots, v_n$ and edges $(v_0, v_1), (v_1, v_2), \cdots, (v_{k-1}, v_k)$.
- if there is a path from u to u', we say u' is *reachable* from u via p
- path is *simple* if all vertices are distinct
- natural notion of *subpath* of p—$\langle v_i, v_{i+1}, \ldots, v_j \rangle$ where $0 \leq i \leq j \leq k$
- in a directed graph, path $\langle v_0, v_1, \ldots, v_k \rangle$ forms a *cycle* if $v_0 = v_k$ and $k \geq 1$
  - cycle is *simple* if $v_1, v_2, \ldots, v_k$ are distinct
  - graph has no self-loops—say it's *simple*
- in an undirected graph, path $\langle v_0, v_1, \ldots, v_k \rangle$ forms a *cycle* if $v_0 = v_k$, $k \geq 3$, and $v_1, v_2, \ldots, v_k$ are distinct
- a graph with no cycles is called *acyclic*

An undirected graph is *connected* if each pair of vertices is connected by a path.

- the *connected components* are the "equivalence classes" of vertices under the "is reachable from" relation

A directed graph is *strongly connected* if every two vertices are reachable from one another.

- the *strongly connected components* of a digraph are the equivalence classes" of vertices under the "are mutually reachable" relation
  - from basic definitions, a vertex always can reach itself (both for directed and undirected)

There is a natural notion of graph *isomorphism*: $G = (V, E)$ is isomorphic to $G' = (V', E')$ if there is a 1-to-1 onto function $f : V \mapsto V'$ such that $(u, v) \in E$ if and only if $(f(u), (f(v)) \in E'$.

- conceptually—can "relabel" G to get G'

The graph $G' = (V', E')$ is a *subgraph* of $G = (V, E)$ if $V' \subset V$ and $E' \subset E$.

- given $v' \subset V$, the *subgraph induced by* V' is $V', (V' \times V') \cap E)$.

Given an undirected graph $G = (V, E)$, the *directed version* of G' is the graph $G' = (V, E')$, where $(u, v) \in E'$ if and only if $(u, v) \in E$.

- conceptually—introduce two edges for each original edge

Given a directed graph $G = (V, E)$, the *directed version* of G' is the graph $G' = (V, E')$, where $(u, v) \in E'$ if and only if $u \neq v$ and $(u, v) \in E$.

- conceptually—remove directionality and self-loops

In a directed graph $G = (V, E)$, a *neighbor* of vertex u is any vertex that is adjacent to u in the undirected version of G. In an undirected graph, u and v are neighbors if they are adjacent.
Special graphs:

1. *complete graph*—undirected graph where every pair of vertices is adjacent
2. *bipartite graph*—undirected graph where the vertex set can be partitioned into two sets V_1 and V_2 such that every edge in the graph is of the form (x, y) where $x \in V_1$ and $y \in V_2$.
3. *forest*—acyclic undirected graph
4. *tree* (sometimes *free tree*)—connected forest
5. *dag*—directed acyclic graph

There are small variations on the notion of a graph—*multigraph*, *hypergraph*.

Trees

CLRS B.5
Many closely related but slightly different definitions.

Free trees

CLRS B.5.1
Free tree—a connected acyclic undirected graph

- usually omit the "free"

Forest—acyclic undirected graph, possibly disconnected

Properties of trees

Theorem 1 Let $G = (V, E)$ be an undirected graph. Then the following are equivalent:

1. G is a free tree
2. any two vertices in G are connected by a unique simple path
3. G is connected but if any edge is removed, the resulting graph is disconnected
4. G is connected and $\mid E \mid = \mid V \mid -1$
5. G is acyclic and $\mid E \mid = \mid V \mid -1$
6. G is acyclic but if any edge is added, the resulting graph is cyclic

Rooted and ordered trees

CLRS B.5.2
A *rooted tree* is a free tree in which one vertex is *distinguished* from the others.

- the distinguished vertex is called the *root*

- vertices in a rooted tree are often called *nodes*

Let r be the root of a rooted tree T. Note that for any node x, there is a unique path from r to x.

- any node y on path from r to x is called an *ancestor* of x
- if y is an ancestor of x, we say x is a *descendant* of y
  - note: each node is an ancestor and a descendant of itself; can talk of *proper* ancestors and descendants
- the *subtree rooted at* x is the tree induced by the descendants of x

If the last edge of the path from r to x is y, we'll say that y is the *parent* of xm and that x is a *child* of y.

- observe—root is only node with no parent
- *siblings*—two nodes which share the same parent
- *external node or leaf*—node with no children
- *internal node* – nonleaf node

Number of children of a node x in a rooted tree T is called the *degree* of x; note the difference in context.
Length of the path from r to x is called the *depth* of x.

- largest depth of any node in T is called the height of T

An *ordered tree* is a rooted tree in which the children at each node are ordered.

Binary and positional trees

CLRS B.5.3

Binary trees are best defined recursively:
A *binary tree* T is a structure defined on a finite set of nodes that either

1. contains no nodes, or

2. is composed of three disjoint sets of nodes: a *root node*, a *left subtree* and a *right subtree*.

The binary tree containing no nodes is called *empty* or *null*.

- will denote by NIL

If the left subtree is nonempty, its root is called the *left child*; similarly define the *right child*.
IMPORTANT—a binary tree is not just an ordered tree in which each node has degree at most two.
A *full binary tree* is a binary tree in which each node is either a leaf or has degree 2.

- generalize to positional tree, k-ary tree, complete k-ary tree
  - how many nodes are there in a complete k-ary tree of height h? how many are leaves? how many internal?