CS 261 – Winter 2010

Trees
Trees

- Ubiquitous – they are everywhere in CS
- Probably ranks third among the most used data structure:
  1. Vectors and Arrays
  2. Lists
  3. Trees
Tree Terminology

• Tree = Set of nodes connected by arcs (or edges)

• A directed tree has a single root node
Tree Terminology

• A *parent* node points to *(one or more)* other nodes

• Nodes pointed to are *children*
Example: Object Taxonomy
Tree Characteristics

- Every node (except the root) has exactly one parent
- Nodes with no children are *leaf* nodes
- Nodes with children are *interior* nodes
Image Representation = Segmentation Tree
Tree Terminology

• Nodes that have the same parent are *siblings*

• The *descendents* of a node consist of its children, and their children, and so on

• All nodes in a tree are descendents of the root (except for the root)
Tree Terminology

• Any node can be considered the root of a *subtree*

• A subtree rooted at a node consists of that node and all of its descendents
Image Representation = Segmentation Tree
Tree Terminology

• There is a single, unique path from the root to any node.

• A path’s *length* is equal to the number of arcs traversed.
Tree Terminology

• A node’s *height* is equal to the maximum path length from that node to a leaf node:
  – A leaf node has a height of 0
  – The height of a tree is equal to the height of the root

• A node’s *depth* is equal to the path length from the root to that node:
  – The root node has a depth of 0
  – A tree’s depth is the maximum depth of all its leaf nodes (which, of course, is equal to the tree’s height)
Example

- Nodes $D$ and $E$ are children of node $B$
- Node $B$ is the parent of nodes $D$ and $E$
- Nodes $B$, $D$, and $E$ are descendents of node $A$
  (as are all other nodes in the tree…except $A$)
- $E$ is an interior node
- $F$ is a leaf node
Tree Characteristics (cont.)

Are these trees?

Yes  No  No