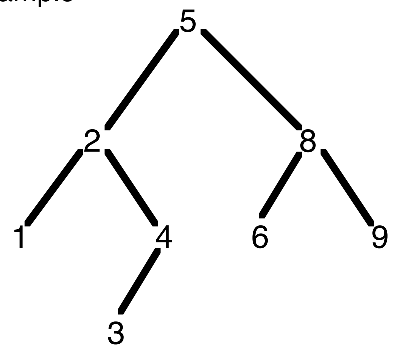
22C:34 — Spring 2004

Binary Search (or Sort) Trees
and Depth First Search

A **BStree** is a rooted tree where
- each node contains (or is) a number,
- the degree of each node is at most 2, and the subtrees of a node (if present) are referred to as *left* and *right* subtrees, and
- for each node of the tree, every item in the left subtree of the node is less than the node, and every item in the right subtree is greater than the node.

For example



Depth First search of binary trees has three variations based on the order of visiting the root node relative to visiting the subtree nodes. These orders are
- *pre-order*– in pre-order, the root node is visited first prior to visiting the nodes of the subtrees,
- *in-order* -- in in-order, the root node is visited between visits to nodes of the left and right subtrees,
- *post-order* – in post-order, the root node is visited last, after visiting the nodes of the subtrees.

An in-order DFS procedure for extracting the items in a BStree into a sorted list is:

```
extract(T) is
  if T = nil
      then return(< >)
      else return (extract(T.Left)^<T.Root>^extract(T.Right)
```