pg 311

Binary Trees

→ an ordered tree w/ following properties:

1.) each node has at most 2 children
2.) each child node is labeled either left or right child
3.) left child precedes a right child in order of children of a node

→ the subtree rooted at a left or right child of an internal node v is called a left subtree or right subtree

→ a binary tree is proper if each node has 0 or 2 children

Recursive Binary Tree Definition

→ node r (root of T which stores an element

→ binary tree called left subtree of T

→ binary tree called right subtree of T

Binary Tree ADT

→ T.left(p): return position p that represents left child of p
→ T.right(p): right child of p
→ T.sibling(p): return position that represents sibling of p

} returns None if empty

Tree Traversal Algorithms:

→ a traversal of T is a systematic way of accessing all positions of T

→ specific action performed upon visiting a node is specific to the algorithm

Preorder/Postorder Traversal

→ in preorder traversal, the root of T is visited first and then the subtrees at its children are traversed recursively

- if the tree is ordered, then the subtrees are traversed according to the order of the children

```
Algorithm preorder(T,p):
    perform the "visit" action for position p
    for each child c in T.children(p)
        preorder(T,c)
```