A Free Tree

Trees

CSE 310 Data Structures and Algorithms

Chapter 13

Section 13.1

Textbook: Chapter 13, Section 13.1

Trees

Check textbook for: (3) \( \Rightarrow \) (2) (1)

Check textbook for: (3) \( \Rightarrow \) (2) (1)

There can be at most one path between any two vertices.

The tree is the graph \( G = (V, E) \) such that:

- \( V \) is a nonempty set of vertices.
- \( E \) is a subset of \( V \times V \) such that:
  - \( \forall u \in V \) there is exactly one vertex \( v \in V \) with \( (v, u) \in E \).
  - \( \forall (u, v), (v, w) \in E \) then \( (u, w) \in E \).

The path obtained by concatenating \( \beta \) and \( \alpha \) is a cycle.
Binary tree \( T \) is

- A non-empty tree
- Has a root node
- Every node has at most two children"
$h = \sum_{i=1}^{Z-1} i = \frac{Z(Z-1)}{2}$

The number of internal nodes is...

* Complete $k$-ary tree
* Complete binary tree
* Complete $Z$-ary tree

Binary Tree Representation

- Each node has a key
- By order with child
  - Node in representation...
Binary-tree traversal: example

- Preorder: visit a node, then traverse its left subtree, and then traverse its right subtree.
- Inorder: traverse the left subtree, visit the node, and then traverse its right subtree.
- Postorder: first traverse the left subtree, traverse the right subtree, and then visit the node.

Assume a pointer to node.

Inorder-Tree-Walk (x)
IF x ≠ Nil
Then Inorder-Tree-Walk(left(x))
print(key(x))
Inorder-Tree-Walk(right(x))

Preorder-Tree-Walk (x)
IF x ≠ Nil
Then print(key(x))
Preorder-Tree-Walk(left(x))
Preorder-Tree-Walk(right(x))

Postorder-Tree-Walk (x)
IF x ≠ Nil
Then Postorder-Tree-Walk(left(x))
Postorder-Tree-Walk(right(x))
print(key(x))

Binary search tree property: search tree in sorted order. Think of binary search.

Node in the right subtree of x, then key > |x|.
If y is a node in the left subtree of x, then key ≥ |x|.
If y is a node in a binary search tree.