

Trees

November 18, 2011

# Previous

- Linear structures
  - Arrays
  - Lists
  - Stacks
  - Queues
- Trees are non-linear

# A Picture & Definitions

Root

Parent

Child

Leaf

Non-leaf

Siblings

Ancestors

Descendants

Subtree

Level

Depth/Height

# Binary Tree

# Quad Tree

- Just so you know that not all trees are binary...

# Tree ADT

- Let's define the **struct Tree**
  - to hold ints

```
typedef struct Tree  
{
```

```
} Tree;
```

# Example Usage

- Pre-fix expression
  - put the operator first
  - $4 + 2$
  - $+ 4 2$
  
- $+ - 2 1 * 9 1$

# Traversals

- inorder: Left, Node, Right
- preorder: Node, Left, Right
- postorder: Left, Right, Node



# Binary Search Tree (BST)

Consider an arbitrary node in a tree called A.

All values in the left subtree are less than the value in A.

All values in the right subtree are greater than the value in A.

# Example

- Insert the following items

100 34 56 99 77 23 1 0 2 98

# Code

- `bstFind(Tree, int)`
- `bstInsert(Tree,int)`
- `bstFindLevel(Tree,int)`
- `bstFindMaxDepth(Tree)`
- `bstDelete(Tree, int)`

# Problems

- What are the main problem with trees?