Previously ...

Binary Trees
Binary Search Trees
definition
search

Today ...

Binary Search Trees
insertion
deletion
traversals

Quiz

Full? Complete? Perfect?

Basic Operations

Do a Search operation
if found, no need to insert (may increase counter)
else, insert node where Search stopped

void BSTree::insert(BSTNode **p, int d) {
    if (!*p) {
        // insert node here
        (*p) = new BSTNode(d);
    } else {
        // call recursively
        if ((*p)->data < d) insert(&((*p)->right), d);
        else if ((*p)->data > d) insert(&((*p)->left), d);
    }
}
void BSTree::insert(int d) {
    insert(&root, d);
}
Remove

Case 1: node is a leaf
trivial, delete node and set parent's pointer to NULL

Case 2: node has 1 child
trivial, set parent's pointer to the only child and delete node

Case 3: node has 2 children
find successor and copy successor's data to node
delete successor

10

11

12

13

14

15

16

17

18
Inorder Traversal

algorithm inorder(p) {
    if (p) {
        inorder(p->left)
        visit(p)
        inorder(p->right)
    }
}

20 21 25 30 40 50 60 75 80

Postorder Traversal

algorithm postorder(p) {
    if (p) {
        postorder(p->left)
        postorder(p->right)
        visit(p)
    }
}

21 25 20 40 30 60 75 80 70 50

How to destroy a binary tree?