

Binary Trees

Lecture 30 Section 12.2

Robb T. Koether

Hampden-Sydney College

Thu, Apr 2, 2009

Outline

Binary Trees

Robb T.
Koether

Introduction

Assignment

1 Introduction

2 Assignment

Binary Trees

Binary Trees

Robb T.
Koether

Introduction

Assignment

Definition (Binary Trees)

A **binary tree** is a data structure with the following properties.

- It is either empty or it has a **root** node.
- Each node in the binary tree may be linked to up to two other nodes, called the **left** and **right children**.
- Each node, except the root node, has exactly one **parent**. The root node has no parent.

A Binary Tree

Binary Trees

Robb T.
Koether

Introduction

Assignment

- The tree metaphor - tree, root, branch, leaf.
- The family metaphor - parent, child, sibling, ancestor, descendant.

Tree Terminology

Binary Trees

Robb T.
Koether

Introduction

Assignment

[Fig30-1.pdf]

Tree Terminology

Binary Trees

Robb T.
Koether

Introduction

Assignment

[Fig30-2.pdf]

Tree Terminology

Binary Trees

Robb T.
Koether

Introduction

Assignment

[Fig30-3.pdf]

Tree Terminology

Binary Trees

Robb T.
Koether

Introduction

Assignment

[Fig30-4.pdf]

Tree Terminology

Binary Trees

Robb T.
Koether

Introduction

Assignment

[Fig30-5.pdf]

Tree Terminology

Binary Trees

Robb T.
Koether

Introduction

Assignment

[Fig30-6.pdf]

Family Terminology

Binary Trees

Robb T.
Koether

Introduction

Assignment

[Fig30-7.pdf]

Family Terminology

Binary Trees

Robb T.
Koether

Introduction

Assignment

[Fig30-8.pdf]

Family Terminology

Binary Trees

Robb T.
Koether

Introduction

Assignment

[Fig30-9.pdf]

Family Terminology

Binary Trees

Robb T.
Koether

Introduction

Assignment

[Fig30-10.pdf]

Family Terminology

Binary Trees

Robb T.
Koether

Introduction

Assignment

[Fig30-11.pdf]

Binary Tree Implementation

Binary Trees

Robb T.
Koether

Introduction

Assignment

Binary Tree Data Member

- `BinaryTreeNode<T>* root` - A pointer to the root node.

Binary Tree Node Data Members

- `T data` - The data value.
- `BinaryTreeNode<T>* left` - A pointer to the left child.
- `BinaryTreeNode<T>* right` - A pointer to the right child.

Insertions and Deletions

Binary Trees

Robb T.
Koether

Introduction

Assignment

- Insertions and deletions in a binary tree are considerably more complicated than they were for linked lists.
- That is because a binary tree is not linear.
- Where should the new node be inserted?
- How would we specify a position?
- When a node is deleted, how is the hole filled?

The `size()` Function

Binary Trees

Robb T.
Koether

Introduction

Assignment

- The `size()` function returns the number of nodes in the tree.
- Since the size of the tree is not stored as a data member, we will need to count the nodes.

The `size()` Function

Binary Trees

Robb T.
Koether

Introduction

Assignment

- How do we count the nodes in a binary tree?
- Think recursively.
- There is the root node.
- There are the nodes in the left subtree.
- There are the nodes in the right subtree.
- Therefore,

$$\text{size}(\text{root}) = 1 + \text{size}(\text{left}) + \text{size}(\text{right}).$$

The `size()` Function

Binary Trees

Robb T.
Koether

Introduction

Assignment

Example (Public, Nonrecursive `size()` Function)

```
int size() const
{
    return size(root);
}
```

The `size()` Function

Binary Trees

Robb T.
Koether

Introduction

Assignment

Example (Private, Recursive `size()` Function)

```
int size(BinaryTreeNode* node) const
{
    if (node == NULL)
        return 0;
    else
        return 1 + size(node->left)
            + size(node->right);
}
```

The `search()` Function

Binary Trees

Robb T.
Koether

Introduction

Assignment

- The `search()` function has prototype

The `search()` Function

```
BinaryTreeNode* search(T value) const;
```

- It returns a pointer to the node where the value was found, or
- It returns `NULL` if the value was not found.

The search() Function

Binary Trees

Robb T.
Koether

Introduction

Assignment

- Write the public and private `search()` functions.

Assignment

Binary Trees

Robb T.
Koether

Introduction

Assignment

Homework

- Read Section x, pages x - x.