

ECE 2574

Introduction to Data Structures and Algorithms

33: Link-Based Binary Trees

Chris Wyatt

Electrical and Computer Engineering

Recall the definition of a Binary Tree

A Binary Tree T is a set of nodes such that

T is empty

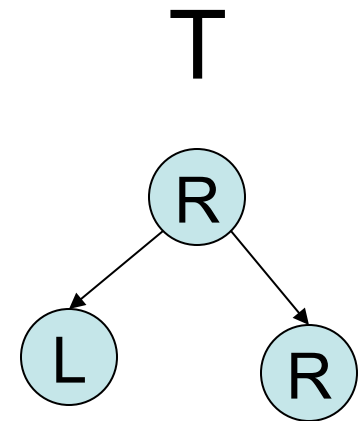
T is partitioned into three subsets:

1. A single node R , the root

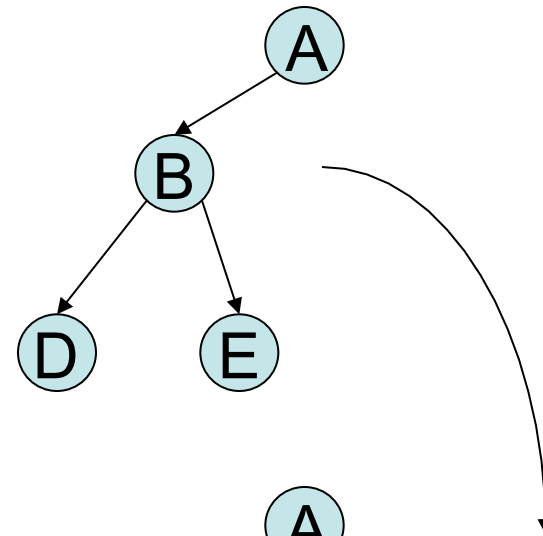
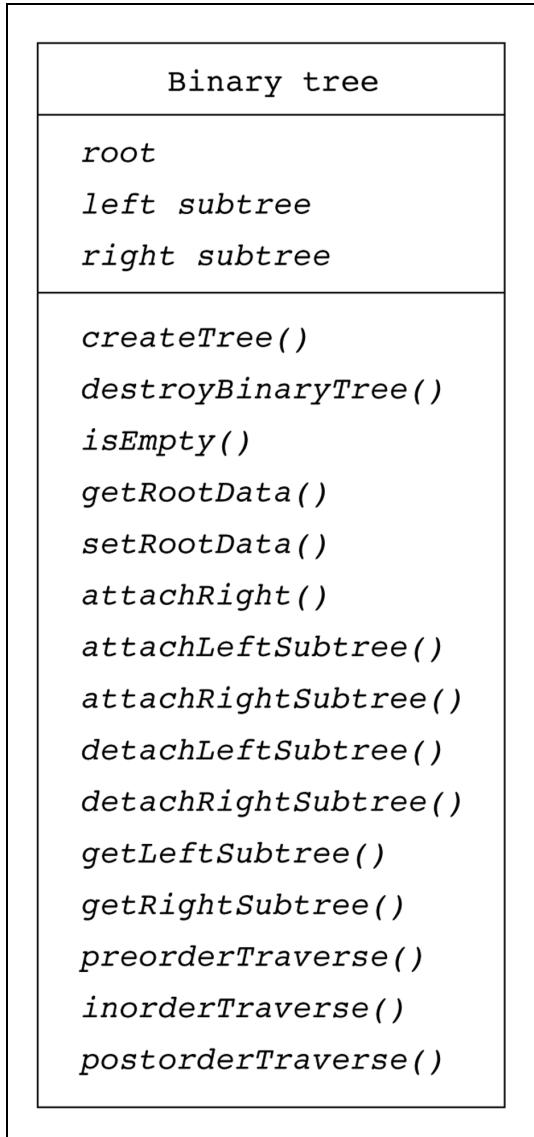
- Two, possible empty sets forming binary trees

2. the left subtree

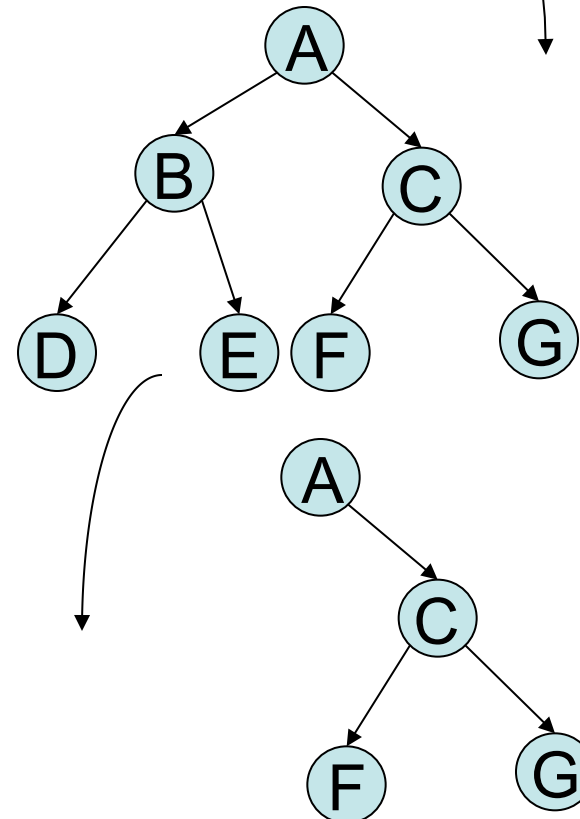
3. the right subtree



The Binary Tree ADT



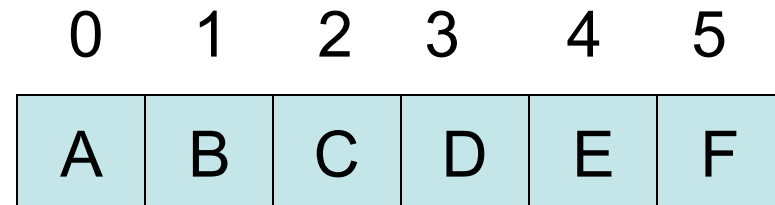
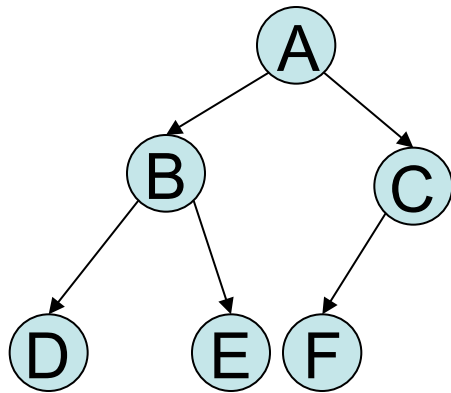
attach right subtree



detach left subtree

Representing Binary Trees

Array based implementation for complete trees

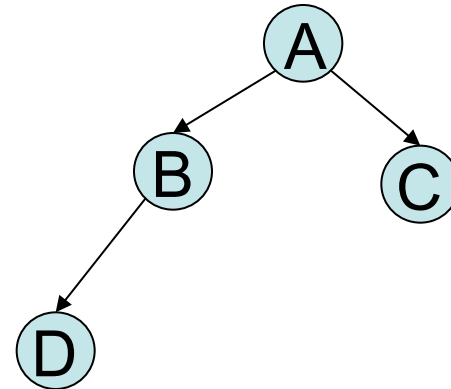


Why does this not work for non-complete trees?

Representing Binary Trees

Pointer based implementation, an extension of a linked list

```
struct node
{
    item a;
    node * left;
    node * right;
}
```



Implementation

See `binarytree/BinaryTree.h` and
`binarytree/BinaryTree.txx`

Based on text source, but templates over the item
type and the visit function.

Recall the Binary Search Tree (BST)

A Binary Search Tree is-a Binary Tree, but with a simple interface.

Insert

Search

Remove

See `bst/binary_search_tree.h` and
`bst/binary_search_tree.txx`

BinarySearchTree
<i>root</i> <i>left subtree</i> <i>right subtree</i>
<i>createBinarySearchTree()</i> <i>destroyBinarySearchTree()</i> <i>isEmpty()</i> <i>searchTreeInsert()</i> <i>searchTreeDelete()</i> <i>searchTreeRetrieve()</i> <i>preorderTraverse()</i> <i>inorderTraverse()</i> <i>postorderTraverse()</i>

Next Actions and Reminders

Read CH pp. 525-543 on the Dictionary ADT
Program 4 due Friday