CS 112 – Introduction to Computing II

Wayne Snyder
Computer Science Department
Boston University

Lecture Eleven

Introduction to Binary Search Trees
Recursive algorithms on BSTs: member, insert
Problems with BSTs: worst case trees

Lab:
Performance of BSTs: average case trees

Next Time:
Recursive algorithms on BSTs: delete
Recursive Tree Traversals

Binary Trees

Linked lists have a single pointer to the next item in a linear sequence:

```
6   13   -4   7   *
```

Recall that ordering a LL does not help much, so we can’t use binary search! The next data structure is an attempt to fix this problem.

**Binary Trees** add an additional pointer, so that the linear sequence becomes an upside-down tree where each node has 0, 1, or 2 branches or “children”:

```
class Node {
    int item;
    Node left;
    Node right;
}
```

Node root;
Binary Trees

Binary Trees are an inherently recursive data structure and best manipulated by recursive algorithms:

**Recursive Definition:**

A Binary Tree is either
- Null (empty tree); or
- A node containing data, with pointers left and right to two Binary Trees

```java
class Node {
    int item;
    Node left;
    Node right;
}
```

Some basic definitions:
- The **root** is the node at the top of the tree.
- Trees under a node are called **subtrees** of that node;
- The **size** of a tree is the number of nodes in it;
- If A points to B, then B is called the **child** of A;
- The **parent** of a node is the (unique) node which points to it (the root has no parent);
- A node is a **leaf node** if it has no children.

Root is A
Size is 7
C is parent of D
Leaf nodes: B, F, G, E
Binary Trees

Some basic definitions:

- A path is a sequence of nodes connected by pointers (from parent to child);
- The length of the path is the number of links;
- If there is a path from A to B of length at least one, then A is an ancestor of B and B is a descendant of A.
- The depth of a node in a tree is the number of links on the path from the root;
- The height of a tree is the maximum depth among any of its nodes (or: the length of the longest path).
- Level K in a tree is all the nodes of depth K.

Path from A to F in red, of length 3
D is descendant of A
C is ancestor of G
Depth of D = 2
Height of tree = 3

In order to use Binary Trees for search, we need to simulate binary search! This requires a different definition (still recursive):

A Binary Search Tree is either
- Null (empty tree); or
- A node containing a key K, with pointers left and right to two Binary Search Trees; all the keys in the left subtree are less than K, and all the keys in the right subtree are greater than K:

This is assuming no duplicates! If duplicates, replace > by >= in right subtree.
Binary Search Trees

We will look at the following algorithms for BSTs on the board:

Size()
Height()
Lookup/member()
Insert()

[next time]
Delete()

Traversal()

Go to: www.cs.bu.edu/fac/snyder/cs112/CourseMaterials/BinaryTreeCode.html