CSE 674 Advanced Data Structures

Trees

Andrew C. Lee

EECS Dept, Syracuse U.

Contents

We will discuss

1. Trees
2. Trees as data structures
3. Tree Traversals
4. Binary Trees
5. Binary Search Trees

Trees

1. Intuitive description: A branch structure between nodes
2. Many variations of trees (sometimes distinctions between them are subtle)
   Reading: CLRS Appendix B.5
3. Free Trees: connected undirected graphs with no cycles
4. In computer science, we like to represent hierarchical structures via *rooted* trees

Rooted Trees

1. Define via recursion

   Definition a tree is a finite set T of one or more nodes where one element, called the root of T, is distinguished. The remaining nodes in the tree form m disjoint subset $(T_1, \ldots, T_m)$ $(m \geq 0)$ where each subset (referred as subtree) is itself a tree.

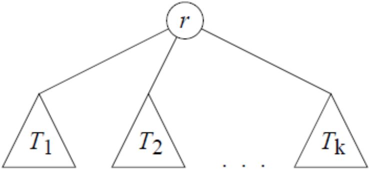
2. Some authors allow the possibly of an *empty* tree with no node.
3. If there is an order among the T_i's, then we will call T an ordered tree.

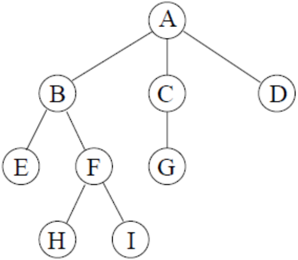
Some Definitions

For an *ordered tree*, we have the following definitions:

1. The *degree of a node n*:
   it is the number of children of n. Note that a leaf node has degree 0
2. A *path* between two nodes n and m:
   it is a sequence of nodes $n = u_1, u_2 \ldots, u_{k-1} u_k = m$ where u_i is a parent of u_{i+1} and in this case, the length of the path is $k - 1$.
3. The *depth of a node n*:
   it is the length of the *unique path* from the root to that node.
4. The *height* of a node n:
   it is the length of the longest path from n to a leaf.
5. The *level* of a node is its depth in the tree.

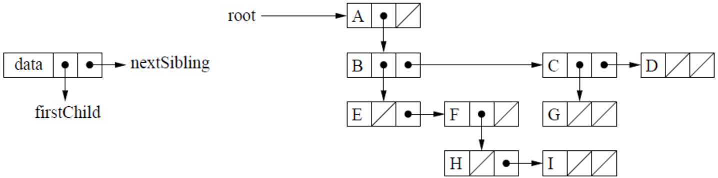
Trees as Data Structures: I





Trees as Data Structures: II

One possible way to implement a rooted tree is:



Question Can you suggest another data structure representation ?

Binary Trees

Definition a binary tree is a structure defined on a finite set of nodes that either

- Contains no nodes, or
- is composed of three disjoint sets of nodes:
  a root node, a binary tree called its left subtree, and a binary tree called its right subtree

Note A heap is a special type of binary trees

Question Suggest a data structure to represent a binary tree

Basic operations on Trees

Many basic operations can be casted as a traversal of nodes in the tree

Examples: Let T be a tree with root r and subtrees $T_1, \ldots, T_m$ ($m \geq 0$)

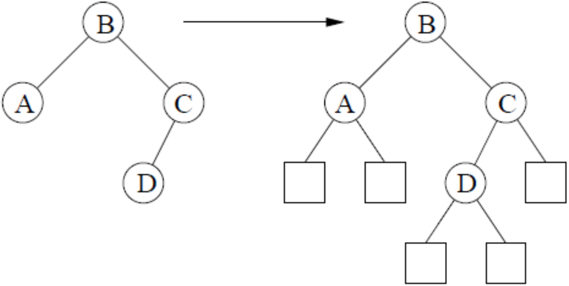
1. Pre-order traversal: visit the root first, then recursively apply pre-order traversal to $T_1, \ldots, T_k$
2. In-order traversal (for binary trees): apply in-order to T_L, visit r and then apply in-order to T_R
3. Post-order traversal: apply post-order traversal to $T_1, \ldots, T_k$ and then visit the root

More on Binary Trees I

1. Extended Binary Trees: replace each null pointer (missing child) with a special leaf node (external node)
2. Threaded Binary Trees: use the null pointer as follows:
   - If a left child pointer of a node n points to null, reset it s.t. it will point to the inorder predecessor of n
   - If a right child pointer of a node n points to null, reset it s.t. it will point to the inorder sucessor of n

   Why are these good ideas ?

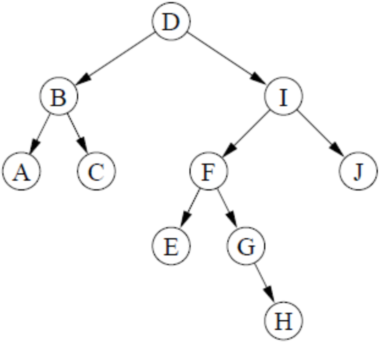
More on Binary Trees II: Extended Binary Trees



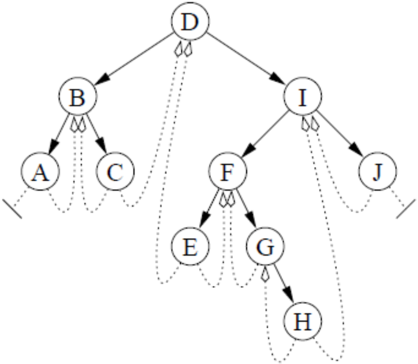
Binary tree and extended binary tree

Note: An extended binary trees with n internal nodes has $n+1$ external nodes.

More on Binary Trees III: Threaded Binary Trees



Binary tree



Binary tree with threads

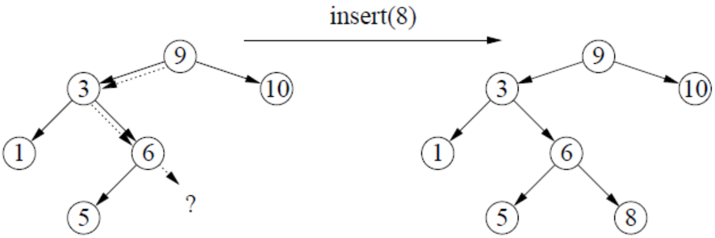
Binary Search Trees

1. Question: What are the drawbacks on using binary search via sequential allocation ?
2. Does using a tree structure helps ?

Operations on Binary Search Trees I

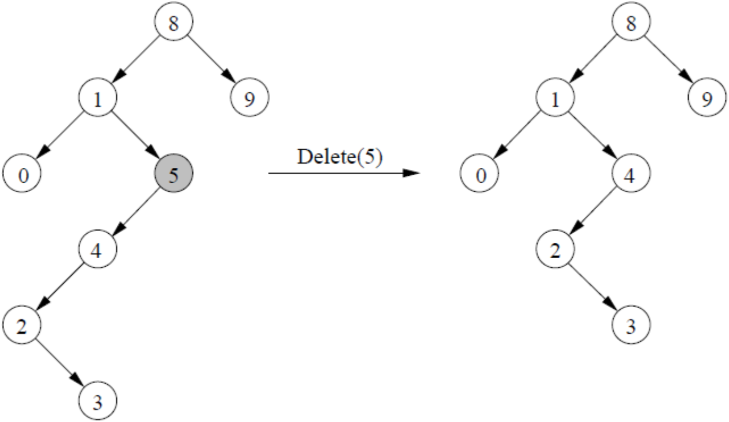
We will discuss how to implement the insert, delete and search operations in class via diagrams.

Insertion:



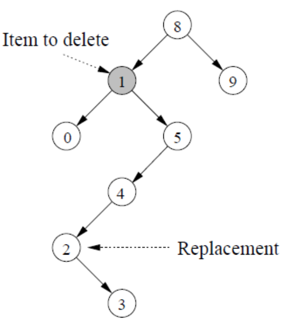
Operations on Binary Search Trees II

Deletion: Case 1



Operations on Binary Search Trees III

Deletion: Case 2



Find the replacement

Delete the replacement

Fill in the hole