CS122 Algorithms and Data Structures

MW 11:00 am - 12:15 pm, MSEC 101

Instructor: Xiao Qin

Lecture 9: Binary Trees

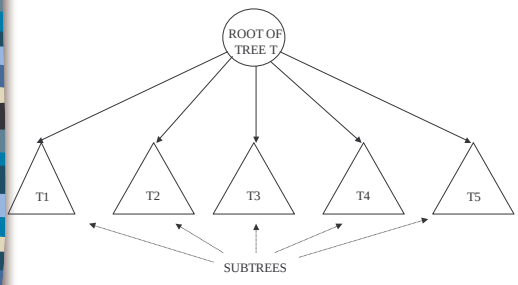
Why Trees?

- Limitations of
  - Arrays
  - Linked lists
  - Stacks
  - Queues

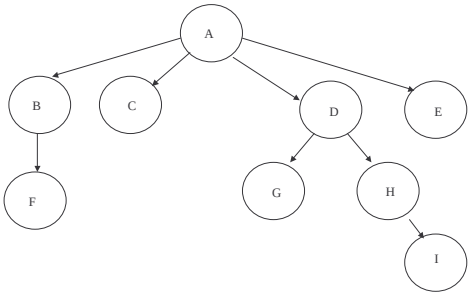
Trees: Recursive Definition

- A tree is a collection of nodes.
- The collection can be empty, or consist of a "root" node *R*.
- There is a "directed edge" from *R* to the root of each subtree. The root of each subtree is a "child" of *R*. *R* is the "parent" of each subtree root.

Trees: Recursive Definition (cont.)



Trees: An Example



Trees: More Definitions

- Nodes with no children are leaves: (C,E,F,H,I).
- Nodes with the same parents are siblings: (B,C,D,E) and (G,H).
- A path from node *n* to node *m* is the sequence of directed edges from *n* to *m*.
- A length of a path is the number of edges in the path

Trees: More Definitions (cont.)

- The level/depth of node *n* is the length of the path from the root to *n*. The level of the root is 0.
- The height/depth of a tree is equal to the maximum level of a node in the tree.
- The height of a node *n* is the length of the longest path from *n* to a leaf. The height of a leaf node is 0.
- The height of a tree is equal to the height of the root node.

Binary Trees – A Informal Definition

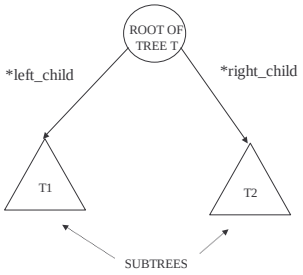
- A binary tree is a tree in which no node can have more than two children.
- Each node has 0, 1, or 2 children
  - In this case we can keep direct links to the children:

```
struct TreeNode
{
    Object  element;
    TreeNode *left_child;
    TreeNode *right_child;
};
```

Binary Trees – Formal Definition

- A binary tree is a structure that
  - contains no nodes, or
  - is comprised of three disjoint sets of nodes:
    - a **root**
    - a binary tree called its **left subtree**
    - a binary tree called its **right subtree**
- A binary tree that contains no nodes is called **empty**

Binary Trees: Recursive Definition



Differences Between A Tree & A Binary Tree

- No node in a binary tree may have more than 2 children, whereas there is no limit on the number of children of a node in a tree.
- The subtrees of a binary tree are ordered; those of a tree are not ordered.

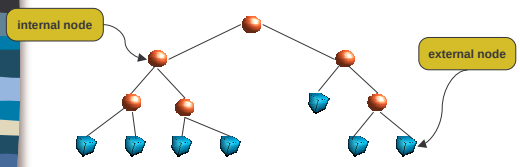
Differences Between A Tree & A Binary Tree (cont.)

- The subtrees of a binary tree are ordered; those of a tree are not ordered

- Are different when viewed as binary trees
- Are the same when viewed as trees

Internal and External Nodes

- Because in a binary tree all the nodes must have the same number of children we are forced to change the concepts slightly
  - We say that all internal nodes have two children
  - External nodes have no children

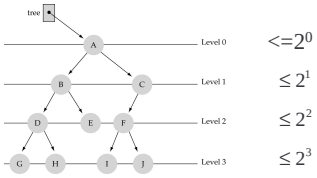


Recursive definition of a Binary Tree

- Most of concepts related to binary trees can be explained recursive
- For instance, A binary tree is:
  - An external node, or
  - An internal node connected to a left binary tree and a right binary tree (called left and right subtrees)
- In programming terms we can see that our definition for a linked list (singly) can be modified to have two links from each node instead of one.

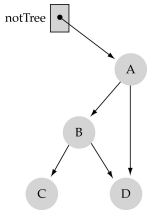
What is a binary tree?

- *Property1*: each node can have up to two successor nodes (*children*)
  - The predecessor node of a node is called its parent
  - The "beginning" node is called the root (no parent)
  - A node without children is called a leaf



What is a binary tree? (cont.)

- *Property2*: a unique path exists from the root to every other node

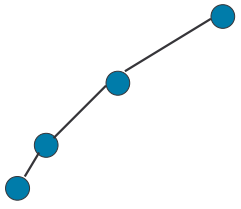


Mathematical Properties of Binary Trees

- Let's us look at some important mathematical properties of binary trees
- A good understanding of these properties will help the understanding of the performance of algorithms that process trees
- Some of the properties we'll describe relate also the structural properties of these trees. This is the case because performance characteristics of many algorithms depend on these structural properties and not only the number of nodes.

Minimum Number Of Nodes

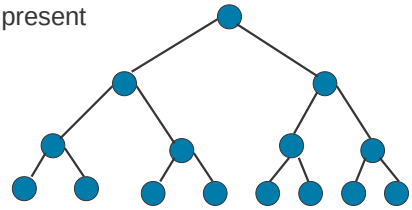
- Minimum number of nodes in a binary tree whose height is h.
- At least one node at each level.



minimum number of nodes is h + 1

Maximum Number Of Nodes

- All possible nodes at first h levels are present



  - Maximum number of nodes

    $= 1 + 2 + 4 + 8 + \ldots + 2^h = 2^{h+1} - 1$

Number Of Nodes & Height

- Let n be the number of nodes in a binary tree whose height is h.
- $h + 1 <= n <= 2^{h+1} - 1$
- $\log_2(n+1)-1 <= h <= n -1$
- The max height of a tree with *N* nodes is *N - 1* (same as a linked list)
- The min height of a tree with *N* nodes is *log(N+1)-1*

Relationship Between Number of Nodes (Internal - External)

- A binary tree with N internal nodes has N+1 external nodes

Let's try to prove this using induction...

Number of edges

- A binary tree with N internal nodes has 2N edges

Let's try to prove this using induction...

Number of edges

- A binary tree with N nodes (internal and external) has N-1 edges

Let's try to prove this using induction...

Binary Tree Representation

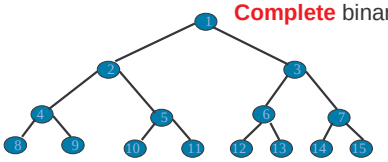
- Array representation
- Linked representation

Binary Trees

- **Full** binary tree :
  - All internal nodes have two children.
- **Complete** binary tree :
  - All leaves have the same level
  - All internal nodes have two children

Node Number Properties

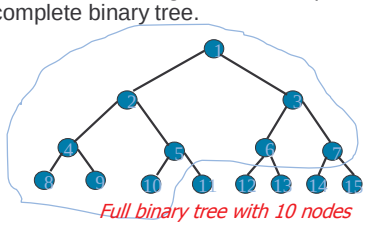
Complete binary tree



- Parent of node i is node i/2
  - But node 1 is the root and has no parent
- Left child of node i is node 2i
  - But if 2i > n, node i has no left child
- Right child of node i is node 2i+1
  - But if 2i+1 > n, node i has no right child

Full Binary Tree With n Nodes

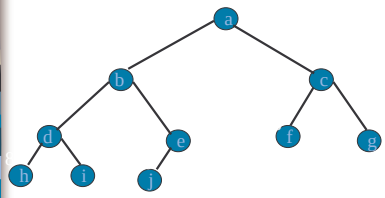
- Start with a full binary tree that has at least n nodes.
- Number the nodes as described earlier.
- The binary tree defined by the nodes numbered 1 through n is the unique n node complete binary tree.



Full binary tree with 10 nodes

Array Representation

- Number the nodes using the numbering scheme for a full binary tree
- Store the node numbered i in tree[i]

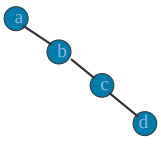


tree[ ]

	a	b	c	d	e	f	g	h	i	j

Right-Skewed Binary Tree

- An n node binary tree needs an array whose length is between n+1 and 2n



tree[ ]

	a	-	b	-	-	-	c	-	-	-	-	-	-	d

Linked Representation

- Each tree node is represented as an object whose data type is `TreeNode`
- The space required by an n node binary tree is n * (space required by one node)

Trees: Linked representation
Implementation 1

```
struct TreeNode
{
    Object  element;
    TreeNode *child1;
    TreeNode *child2;
    .
    .
    .
    TreeNode *childn;
};
```

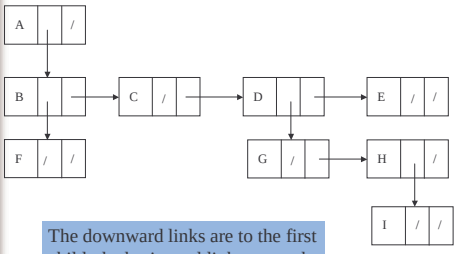
- Each node contains a link to all of its children.
- This isn't a good idea, because a node can have an arbitrary number of children!

Trees: Linked representation
Implementation 2

```
struct TreeNode
{
    Object  element;
    TreeNode *child1;
    TreeNode *sibling;
};
```

Each node contain a link to its first child and a link to its next sibling. This is a better idea.

Implementation 2: Example



The downward links are to the first child; the horizontal links are to the next sibling.

Binary Trees

- A ***binary tree*** is a tree whose nodes have at most two offspring
- **Example**

```
struct nodeType {
    object element;
    struct nodeType *left, *right;
};
struct nodeType *tree;
```

Linked Representation Example