

CS122 Algorithms and Data Structures

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

Instructor: Xiao Qin

Lecture 11: Binary Tree Traversal

Binary Tree Traversal

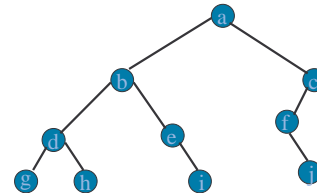
- n Many binary tree operations are done by performing a **traversal** of the binary tree
- n In a traversal, each element of the binary tree is **visited** exactly once
- n During the **visit** of an element, all action (make a clone, display, evaluate the operator, etc.) with respect to this element is taken

Binary Tree - Traversal

- n Traversal of a binary tree is defined recursively. For example,
 1. Visit the node (root or subroot)
 2. Traverse the left subtree
 3. Traverse the right subtree

NOTE there is no correct order. It is possible to perform the above 3 steps in **any (!) order**.

Traversal Applications



- § Make a clone
- § Determine height
- § Determine number of nodes

Traversing Trees

- n Depth-first traversal: This strategy consists of searching deeper in the tree whenever possible. Certain depth-first traversals occurs frequently enough that they are given names of their own
- n Breadth-first traversal: This is a very simple idea which consists of visiting the nodes based on their level in the tree. It visits all nodes with depth 0, then depth 1, then depth 2, and so on.

Special cases of depth-first traversal for binary trees

- n In binary trees there are three basic ways to traverse a tree using the a depth-first search idea
 - **Preorder**: We visit a node, then visit the left and the right subtrees
 - **Inorder**: We visit the left subtree then we visit the node, then we visit the right subtree
 - **Postorder**: We visit the left and right subtree and then we visit the node. This is what normally authors mean if they mention just depth -first traversal

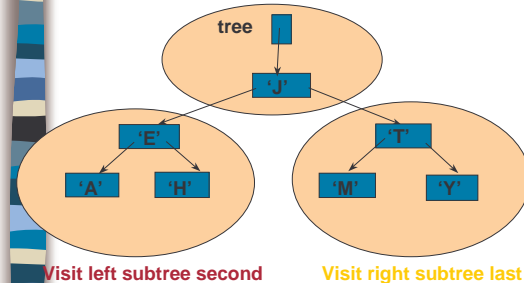
Binary Tree Traversal

Method

- n Preorder
- n Inorder
- n Postorder

- n In a traversal of a binary tree, each element of the binary tree is **visited** exactly once
- n During the **visit** of an element, all action (make a clone, display, evaluate the operator, etc.) with respect to this element is taken

Preorder Traversal: J E A H T M Y



8

Preorder Traversal (cont.)

- n Visit the root of the tree first, then visit the nodes in the left subtree, then visit the nodes in the right subtree

```

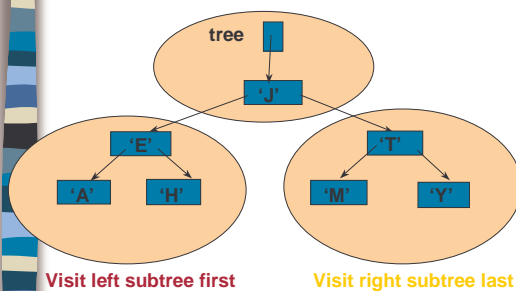
Preorder(tree)
If tree is not NULL {
  Visit Info(tree)
  Preorder(Left(tree))
  Preorder(Right(tree))
}
  
```

Preorder Traversal (cont.)

```

void preOrder(BinaryTreeNode t)
{
  if (t != null)
  {
    visit(t);
    preOrder(t.leftChild);
    preOrder(t.rightChild);
  }
}
  
```

Inorder Traversal: A E H J M T Y



11

Inorder Traversal (cont.)

- n Visit the nodes in the left subtree, then visit the root of the tree, then visit the nodes in the right subtree

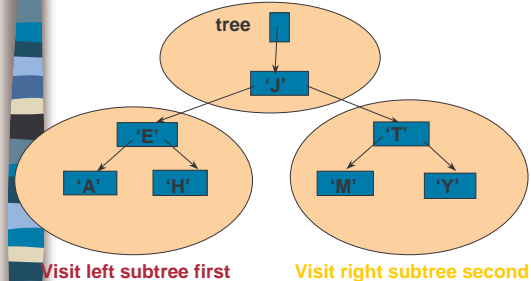
```

Inorder(tree)
If tree is not NULL {
  Inorder(Left(tree))
  Visit Info(tree)
  Inorder(Right(tree))
}
  
```

Inorder Traversal (cont.)

```
void inOrder(BinaryTreeNode t)
{
    if (t != null)
    {
        inOrder(t.leftChild);
        visit(t);
        inOrder(t.rightChild);
    }
}
```

Postorder Traversal: A H E M Y T



14

Postorder Traversal

- n Visit the nodes in the left subtree first, then visit the nodes in the right subtree, then visit the root of the tree

```
Postorder(tree)
If tree is not NULL {
    Postorder(Left(tree))
    Postorder(Right(tree))
    Visit Info(tree)
}
```

Postorder Traversal

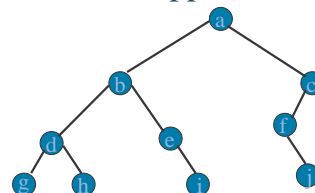
```
void postOrder(BinaryTreeNode t)
{
    if (t != null)
    {
        postOrder(t.leftChild);
        postOrder(t.rightChild);
        visit(t);
    }
}
```

Breadth-first traversal of a tree

- n A breadth-first traversal consists of visiting the nodes based on their level in the tree. It visits all nodes with level depth 0, then depth 1, then depth 2, and so on.

- n Use a queue to implement breadth-first traversal

Traversal Applications



- § Make a clone
- § Determine height
- § Determine number of nodes

Binary Tree Construction

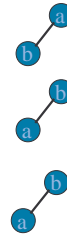
- n Suppose that the elements in a binary tree are distinct
- n Can you construct the binary tree from which a given traversal sequence came?
- n When a traversal sequence has more than one element, the binary tree is not uniquely defined

Some Examples

preorder
r = ab

inorder
= ab

postorder
= ab



Binary Tree Construction

- n Can you construct the binary tree, given two traversal sequences?
- n Depends on which two sequences are given

Preorder And Postorder

preorder = ab
postorder = ba



Preorder and postorder do not uniquely define a binary tree.

Nor do preorder and level order (same example)

Nor do postorder and level order (same example)