Trees

(part 2)

CSE 2320 – Algorithms and Data Structures
Vassilis Athitsos
University of Texas at Arlington

Last updated: 4/19/18

Student Self-Review

- Review the theoretical lecture on trees that was covered earlier in the semester.
- Review your notes and materials on implementing trees in C.

Defining Nodes for Binary Trees

```
typedef struct node *link;
struct node {
  Item item;
  link left;
  link right;
};
```

Other possible fields:

- `link parent;`
- `int size; //`size of subtree rooted at this node. Useful for balancing trees.

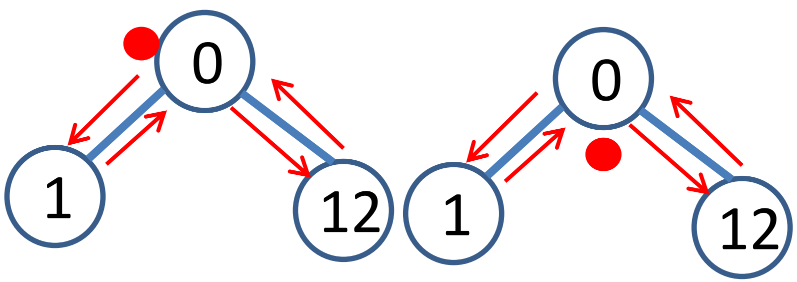
Traversing a Binary Tree

- **Traversing** is the process of going through each node of a tree, and doing something with that node. Examples:
  - We can print the contents of the node.
  - We can change the contents of the node.
  - We can otherwise use the contents of the node in computing something.
- There are four standard choices for the order in which we visit nodes when we traverse a binary tree.
  - **Preorder (Root, L, R)**: we visit the node, then its left subtree, then its right subtree. (depth-first order)
  - **Inorder (L, Root, R)**: we visit the left subtree, then the node, then the right subtree. (depth-first order)
  - **Postorder (L, R, Root)**: we visit the left subtree, then the right subtree, then the node. (depth-first order)
  - **Level order**: all the nodes on the level going from 0 to the last level. (breadth-first)

Examples

(E.g. if printing the nodes, the bullet indicates that you would print at that time.)

preorder inorder postorder



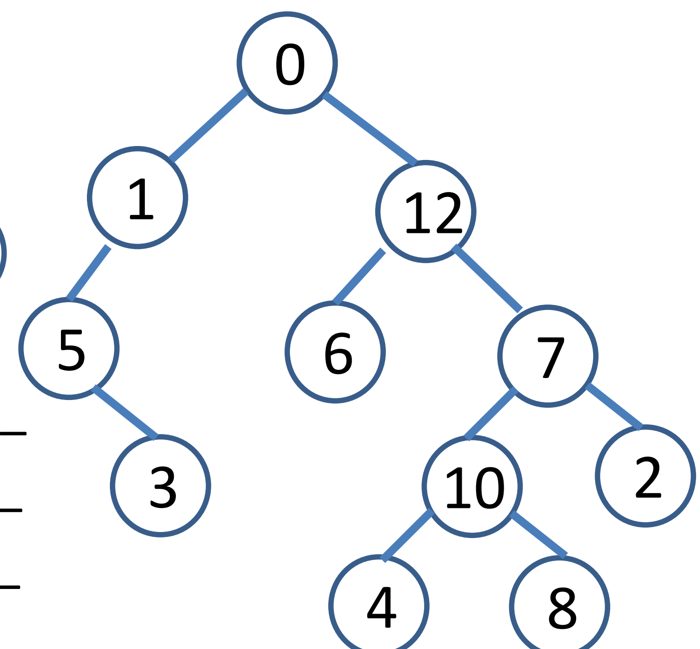
List the nodes of the tree in:

Preorder (____, ____, ____): ________________________________

Inorder (____, ____, ____): ________________________________

Postorder (____, ____, ____): _______________________________

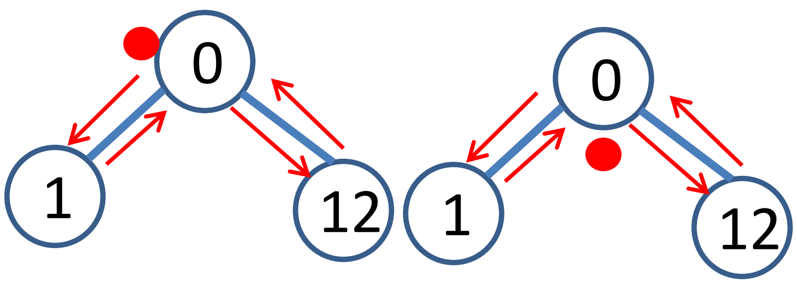
Level-order: _________________________________



Examples

(E.g. if printing the nodes, the bullet indicates that you would print at that time.)

preorder inorder postorder



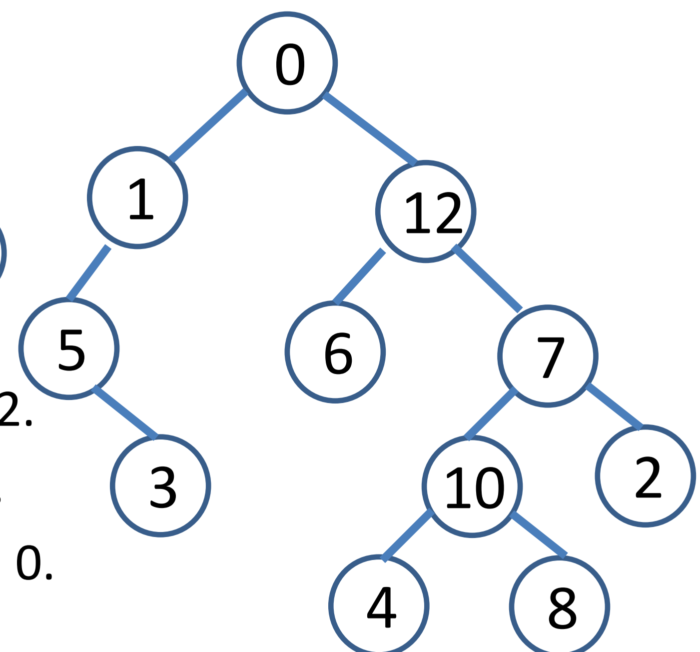
List the nodes of the tree in:

Preorder (Root, Left, Right): 0, 1, 5, 3, 12, 6, 7, 10, 4, 8, 2.

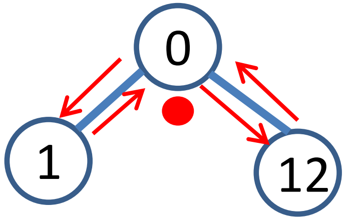
Inorder (Left, Root, Right): 5, 3, 1, 0, 6, 12, 4, 10, 8, 7, 2.

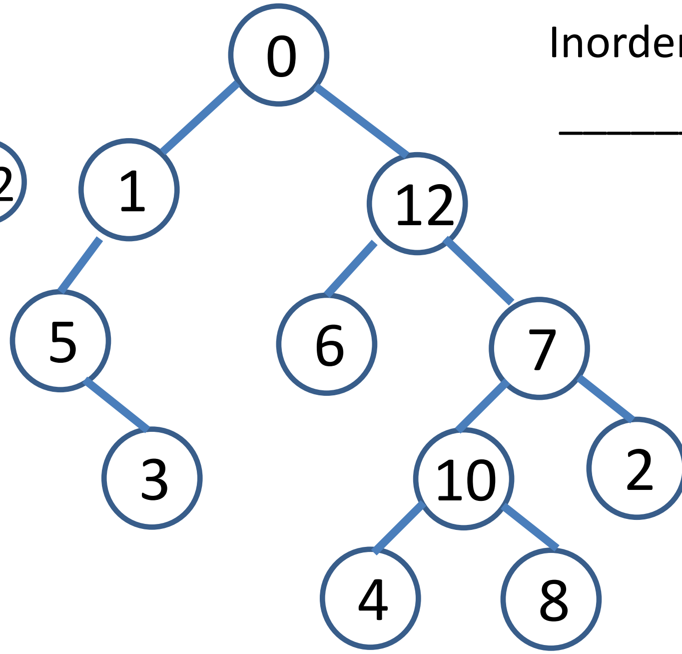
Postorder (Left, Right, Root): 3, 5, 1, 6, 4, 8, 10, 2, 7, 12, 0.

Level-order: 0, 1, 12, 5, 6, 7, 3, 10, 2, 4, 8.



inorder

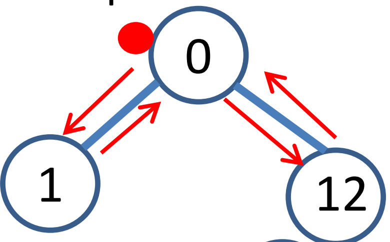




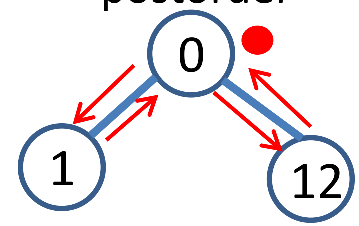
Inorder (____, ____, ____):

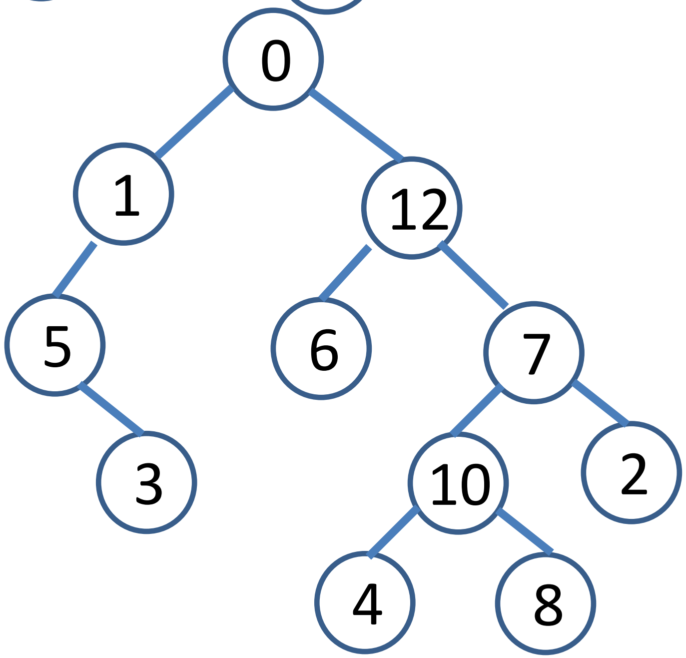
____

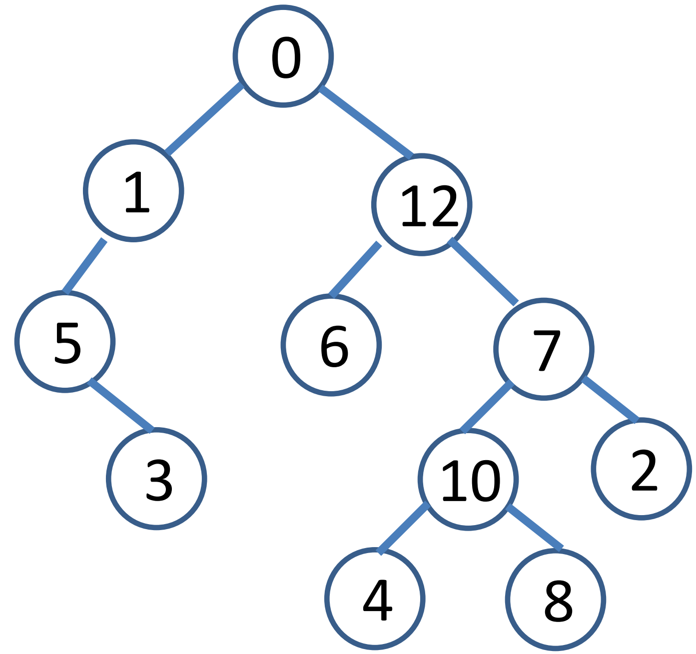
preorder



postorder







Preorder (____, ____, ____):

____

Postorder (____, ____, ____):

____

Recursive Tree Traversal

```
void traverse_preorder(link h){
    if (h == NULL) return;
    do_something_with(h);
    traverse_preorder (h->left);
    traverse_preorder (h->right);
}
```

```
void traverse_inorder(link h){
    if (h == NULL) return;
    traverse_inorder (h->left);
    do_something_with(h);
    traverse_inorder (h->right);
}
```

```
void traverse_postorder(link h){
    if (h == NULL) return;
    traverse_postorder(h->left);
    traverse_postorder(h->right);
    do_something_with(h);
}
```

For a tree with N nodes:

Time complexity:

Space complexity:

Class Practice

- Write the following (recursive or not) functions, in class:
  - Count the number of nodes in a tree
  - Compute the height of a tree
  - Level-order traversal – discuss/implement
  - Print the tree in a tree-like shape – discuss/implement

- Which functions are "similar" to the traversals discussed previously and to each other?

- These slides contain code from the Sedgewick book.

Recursive Examples

Counting the number of nodes in the tree:

```
int count(link h){
    if (h == NULL) return 0;
    int c1 = count(h->left);
    int c2 = count(h->right);
    return c1 + c2 + 1;
}
```

Computing the height of the tree:

```
int height(link h){
    if (h == NULL) return -1;
    int u = height(h->left);
    int v = height(h->right);
    if (u > v)
        return u+1;
    else
        return v+1;
}
```

Recursive Examples: print tree

Print the contents of each node (assuming that the items in the nodes are characters)

How will the output look like?

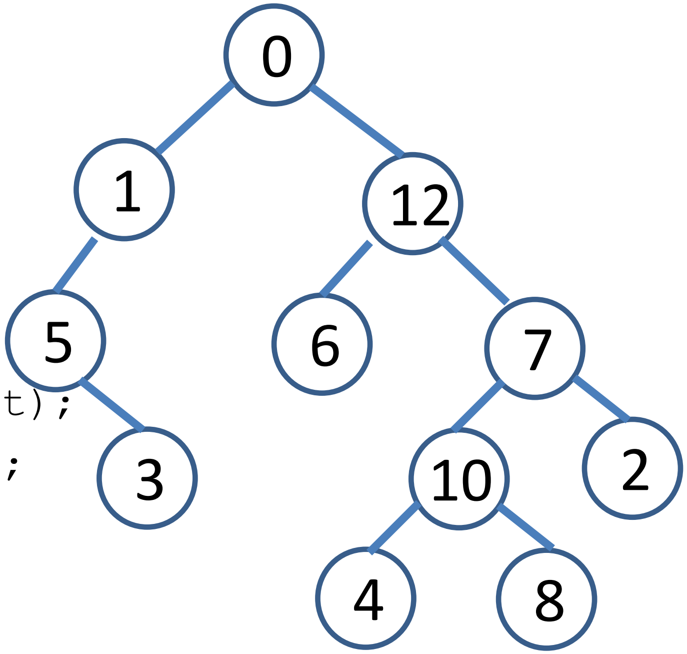
What type of tree traversal is this?

```
void printnode(char c, int h) {
    int i;
    for (i = 0; i < h; i++)
       printf("  ");
    printf("%c\n", c);
}

void show(link x, int h) {
    if (x == NULL) {
       printnode("*", h);
       return;
    }
    printnode(x->item, h);
    show(x->left, h+1);
    show(x->right, h+1);
}
```

Recursive and Iterative Preorder Traversal (Sedgewick)

```
void traverse(link h, void (*visit)(link))  {
    if (h == NULL) return;
    (*visit)(h);
    traverse(h->left, visit);
    traverse(h->right, visit);
  }
-----
void traverse(link h, void (*visit)(link)) {
    STACKinit(max); STACKpush(h);
    while (!STACKempty())
      {
        (*visit)(h = STACKpop());
        if (h->right != NULL) STACKpush(h->right);
        if (h->left != NULL) STACKpush(h->left);
      }
  }
```

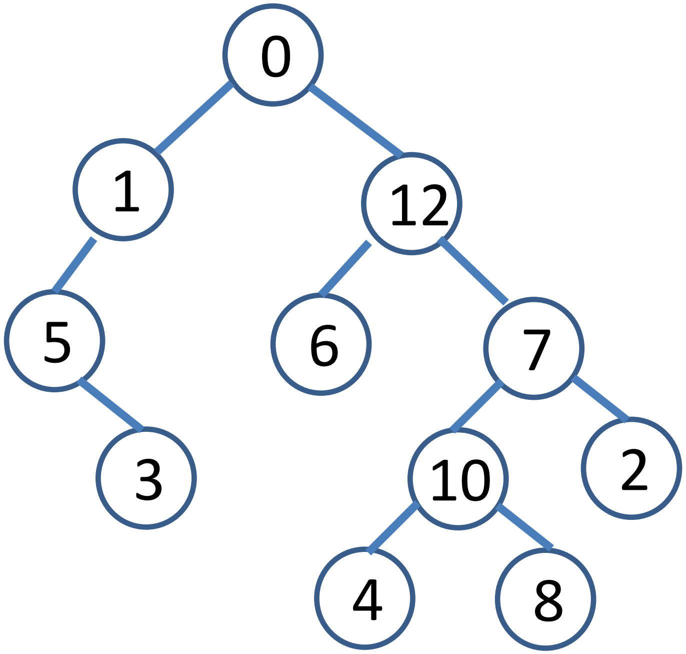


Stack:

Print:

Level-Order Traversal (for printing)

```
// Adapted from Sedgewick
void traverse(link h) {
    Queue Q = new_Queue();
    put(Q,h);
    while (!empty(Q)) {
        h = get(Q); //gets first node
        printItem(h->item);
        if (h->left != NULL) put(Q,h->left);
        if (h->right != NULL) put(Q,h->right);
    }
}
```



Queue:________________________________

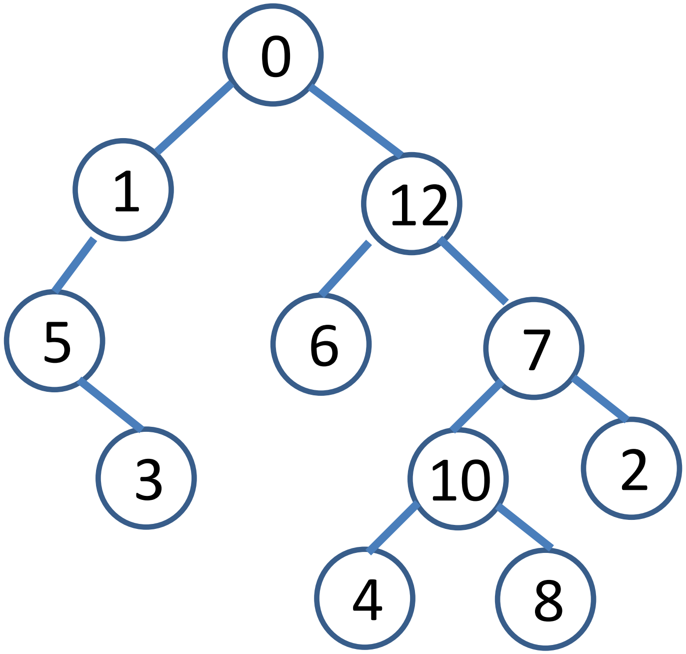
Print:________________________________

Level-Order Traversal
(with function arguments)

```
// Adapted from Sedgewick
void traverse(link h, void (*visit)(link)) {
    Queue Q = new_Queue();
    put(Q,h);
    while (!empty(Q)) {
        (*visit)(h = get(Q)); //gets first node
        if (h->left != NULL) put(Q,h->left);
        if (h->right != NULL) put(Q,h->right);
      }
  }
```

Queue:________________________________

Print:________________________________



General Trees

- In a general tree, a node can have any number of children.

- How would you implement a general tree?

General Trees

- In a general tree, a node can have any number of children.
- Left-child - right-sibling implementation
  - Draw tree and show example
  - (There is a one-to-one correspondence between ordered trees and binary trees)