CODING BLOCKS

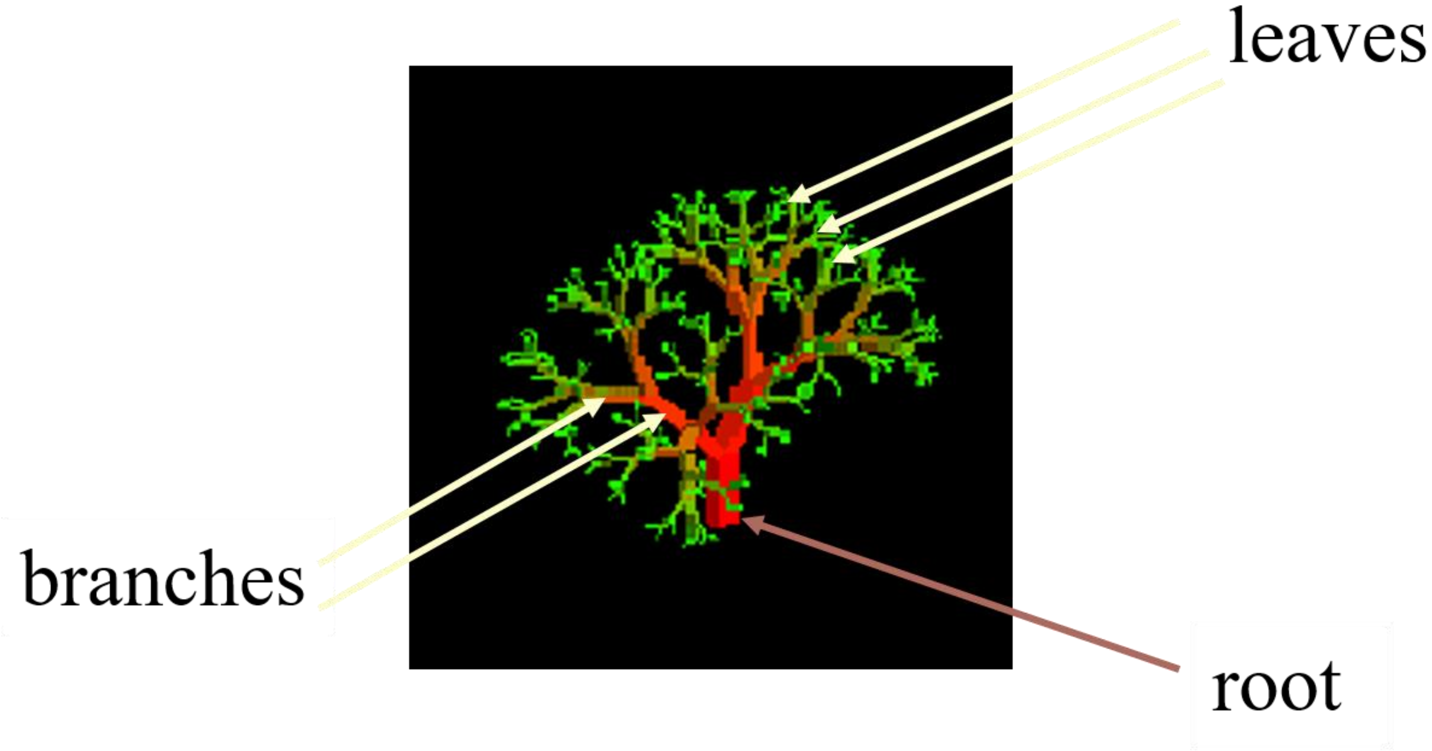
Lecture-15

Data Structures

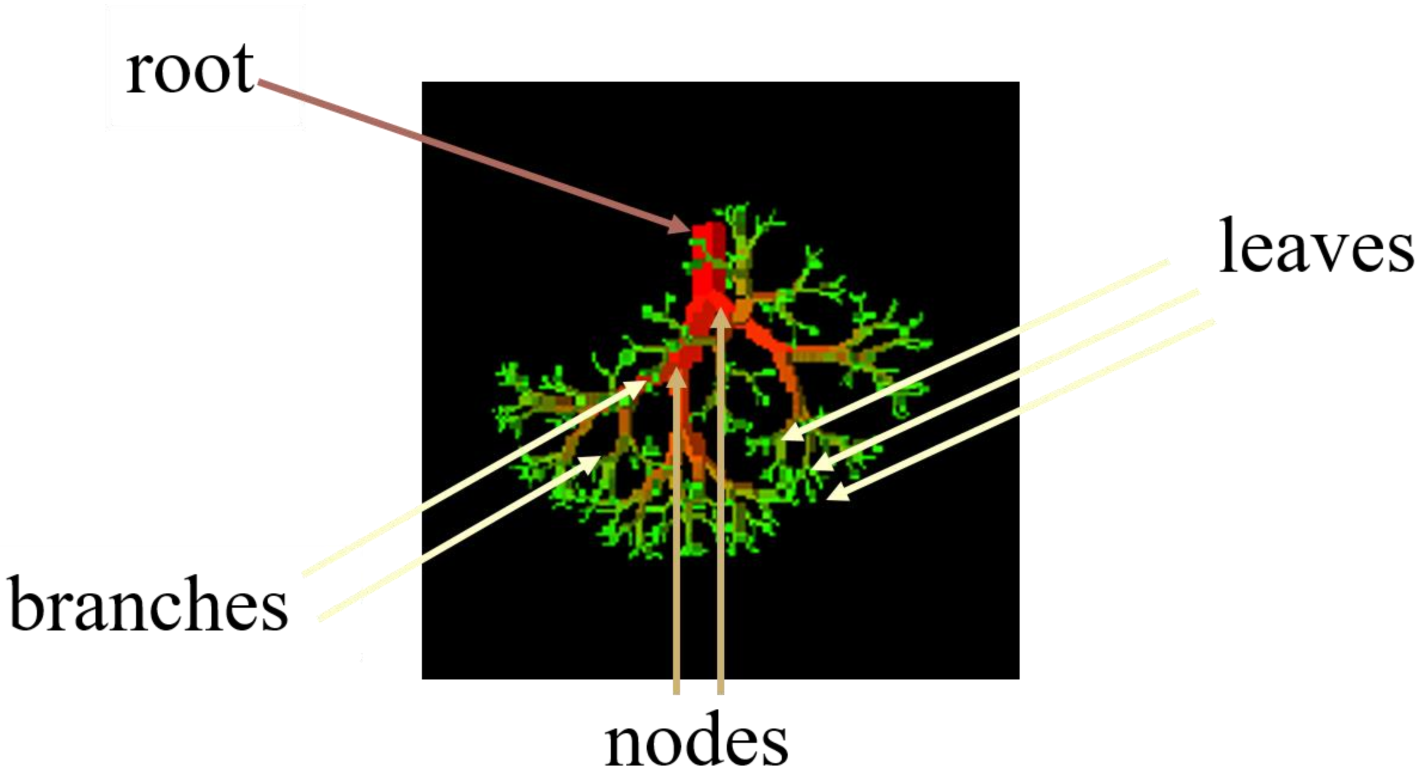
- Generic Trees

Deepak Aggarwal
deepak@codingblocks.com

Nature View of a Tree

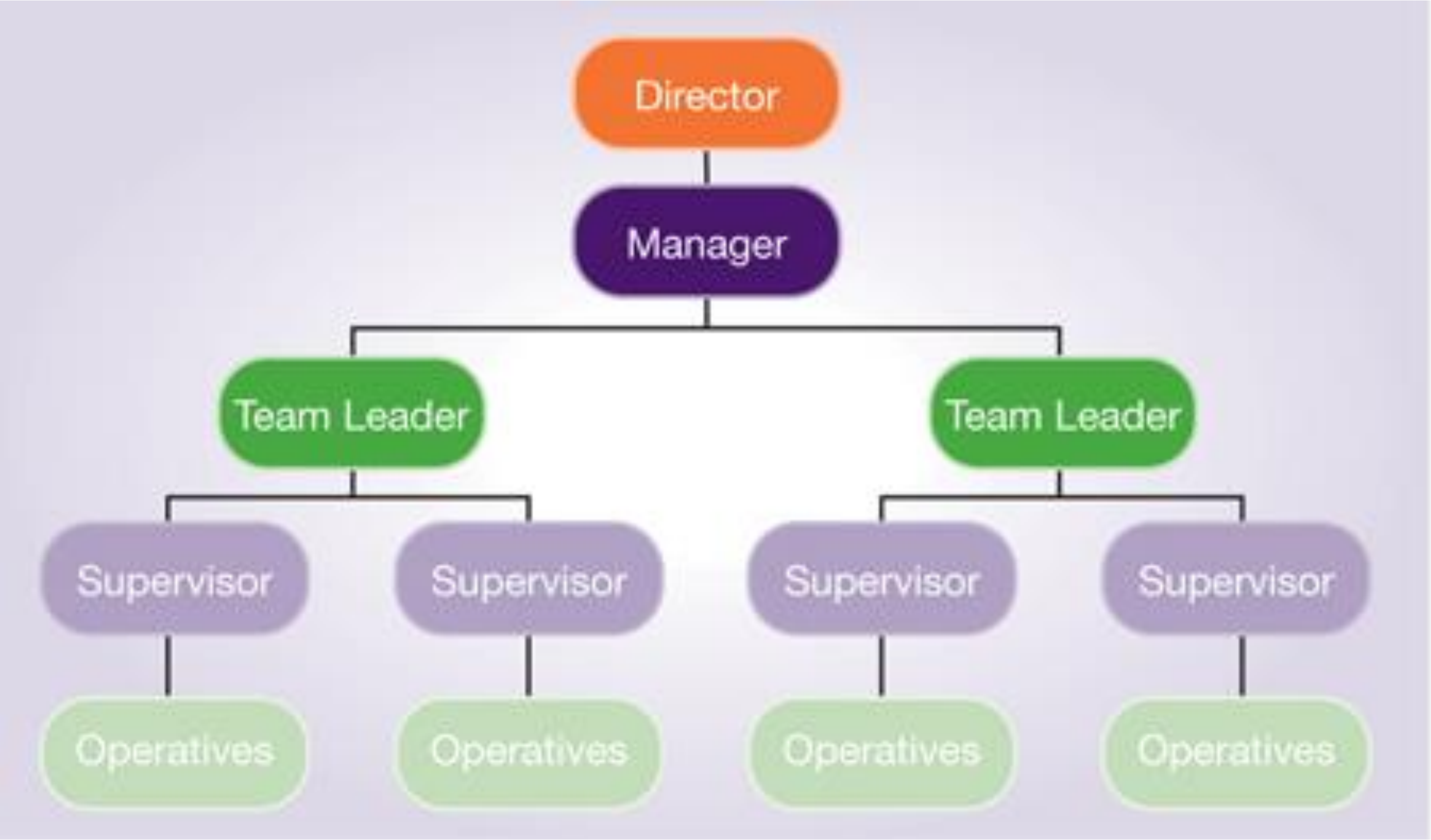


Computer Scientist's View



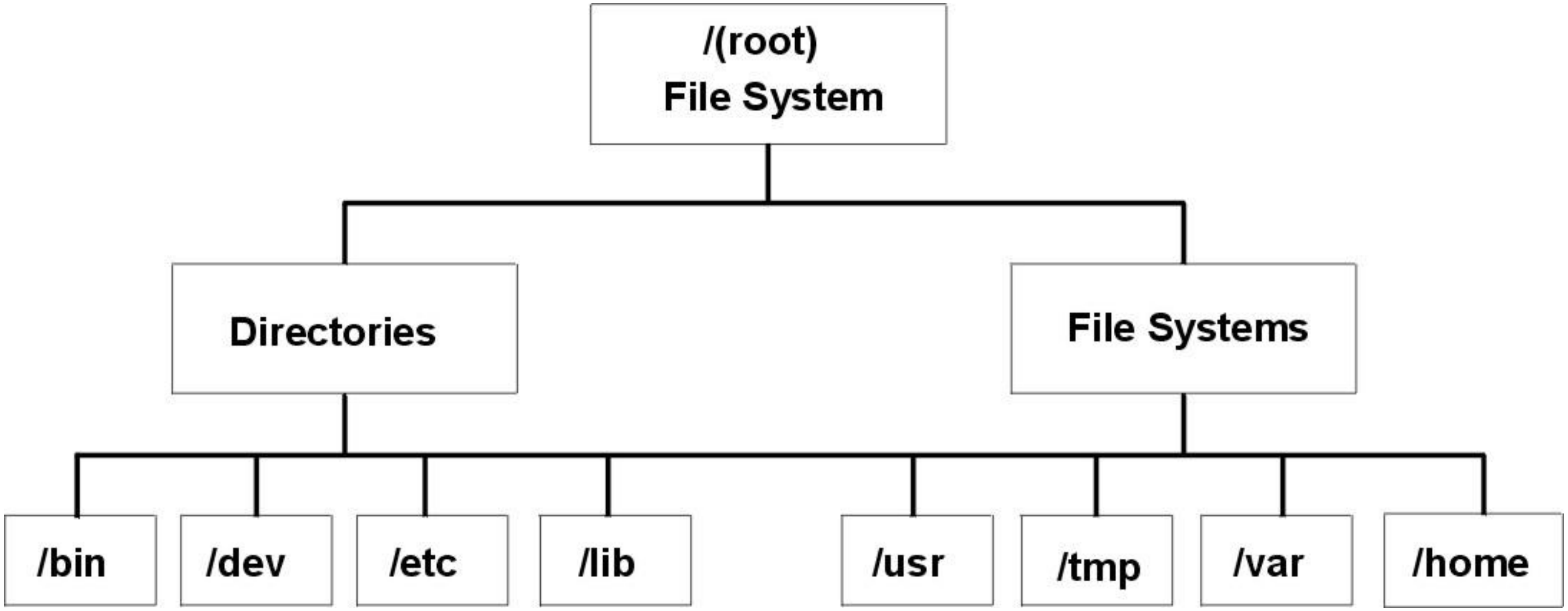
CODING BLOCKS

Organization Hierarchy



CODING BLOCKS

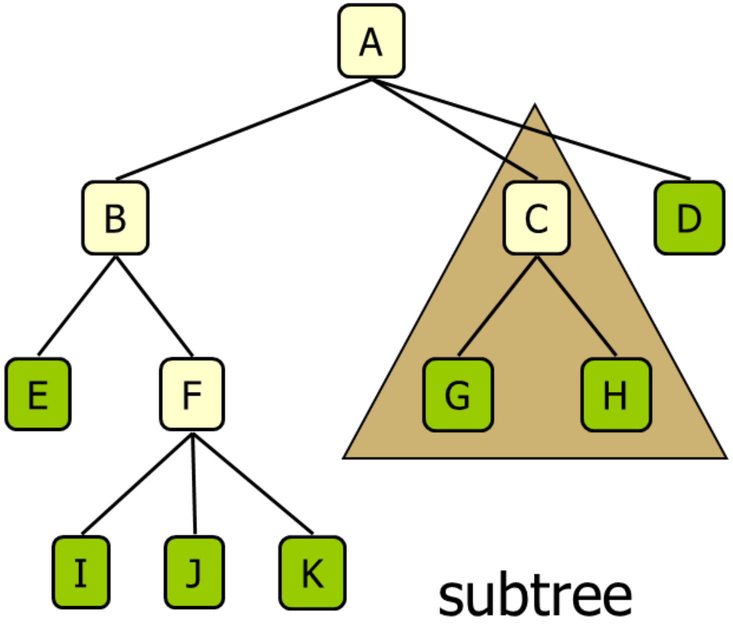
UNIX File System



CODING BLOCKS

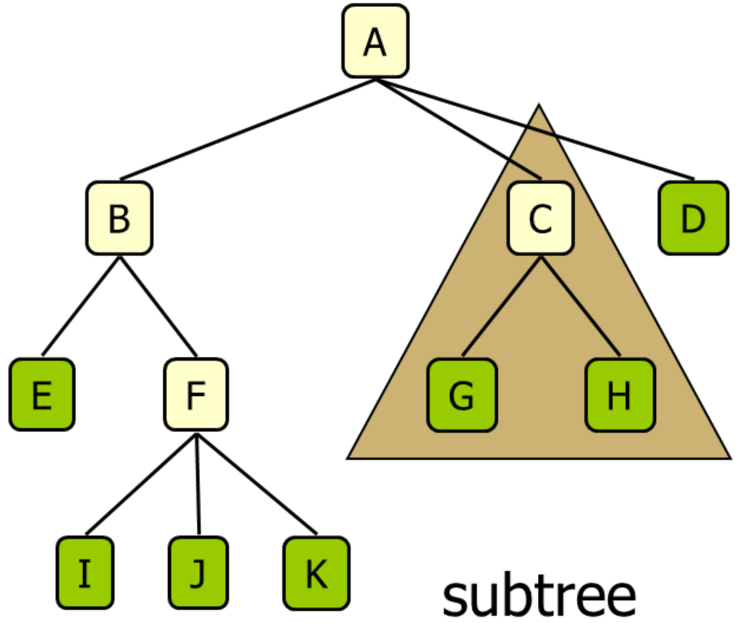
Tree Terminologies

1. **Node** of Tree
2. **Root** of Tree
3. **Children** of Node
4. **Parent** of Node
5. **Ancestor** of Node
6. **Descendants** of Node
7. **Sibling** of Node
8. **Leaves** of Tree



Tree Important Properties

1. **Degree** of a Node
2. **Degree** of Tree
3. **Depth** of a Node
4. **Height** of Tree
5. **Subtree**



How to Implement a Node of a Tree

```
class node{
    int data;
    node* children[];
    int children_count;
    node* parent; //Optional
}
```

How to Implement a Tree

Use Nodes to create tree in every program

Lets see how to input and output Tree

1. Write a function to take tree as input from user
2. Print out a tree

Lets discuss few problems

1. Find the node with largest data in a tree
2. Print all the elements at depth K.

Your Turn

1. Find number of Nodes greater than an integer x
2. Find the node for which sum of the data of all children and the node itself is maximum

A tree walk or traversal is a way of visiting all the nodes in a tree in a specified order.

Lets code these tree traversals

1. Preorder Traversal(Recursive)
2. Preorder Traversal(Iterative)
3. Postorder Traversal
4. Levelorder Traversal

Our Tree class

```
Class Tree {
    node* root;
    int size();
    boolean isEmpty();
    node* root();
    node* parent(node*);
    node** children(node*); // etc etc
}
```

CODING BLOCKS

Thank You!

Deepak Aggarwal