

Data Structures & Algorithms - I

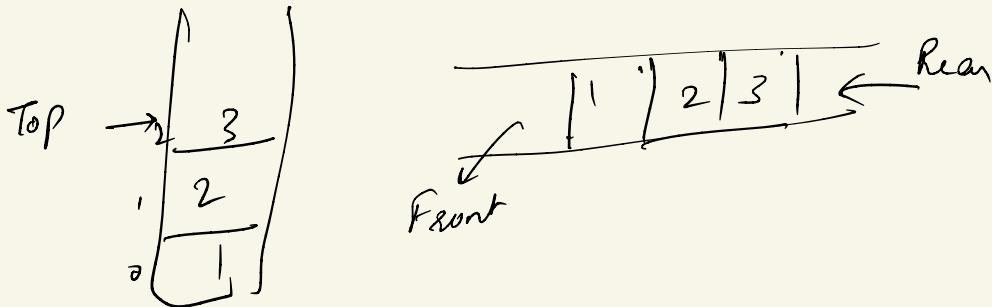
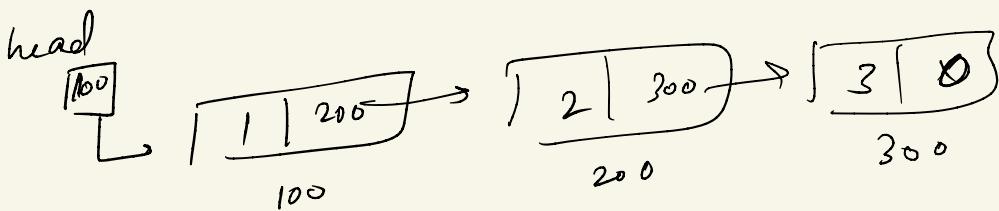
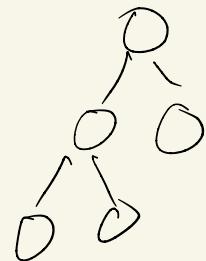
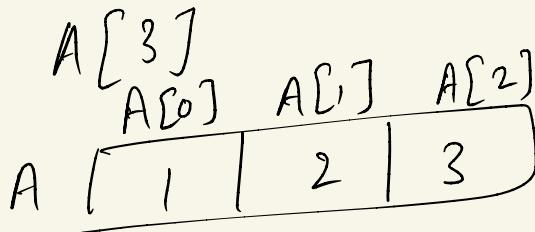
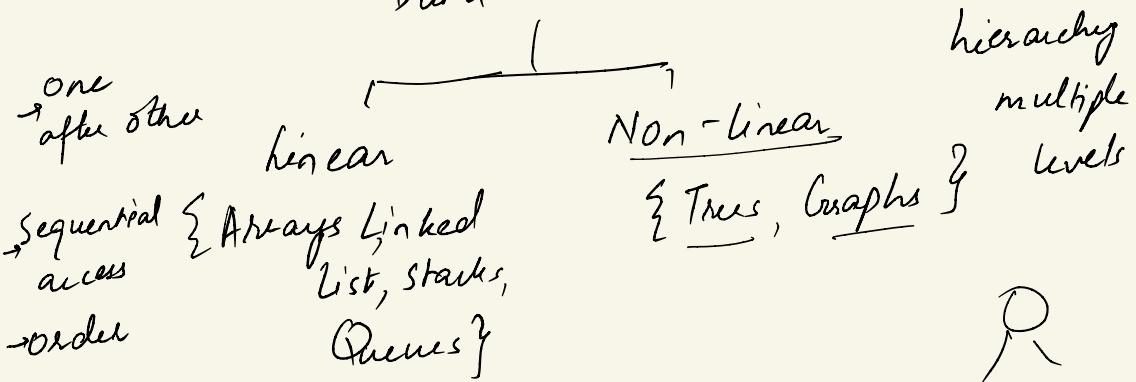
Lecture on 01-04-2020

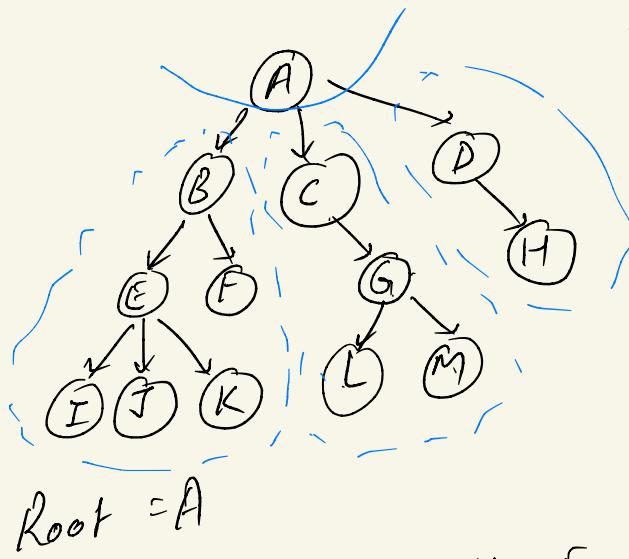
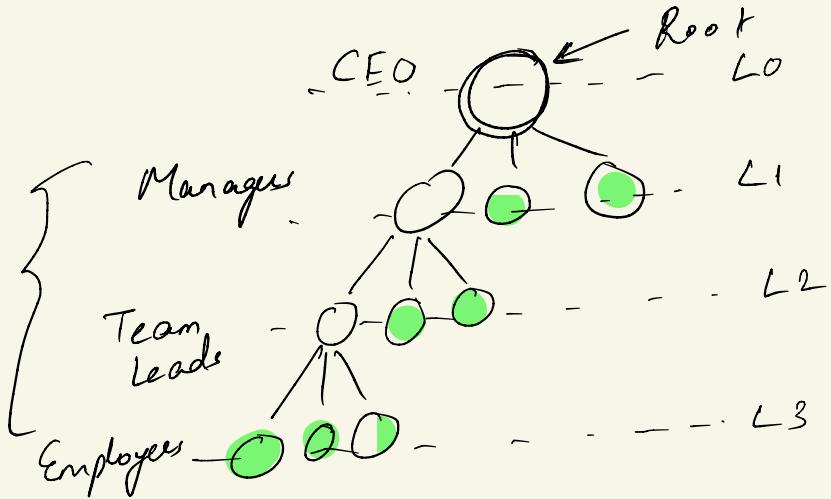
Dr. Prerana Mukherjee



Introduction on Trees

Data Structures





Tree can be defined as a collection of entities (nodes) to simulate some form of hierarchy subtree

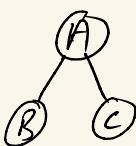
External / Leaf nodes = I, J, K, F, L, M, H

Internal / Non-leaf

Intermediate = B, C, D, E, G

parent
ancestor

child
descendent



degree of node = no of children at that node

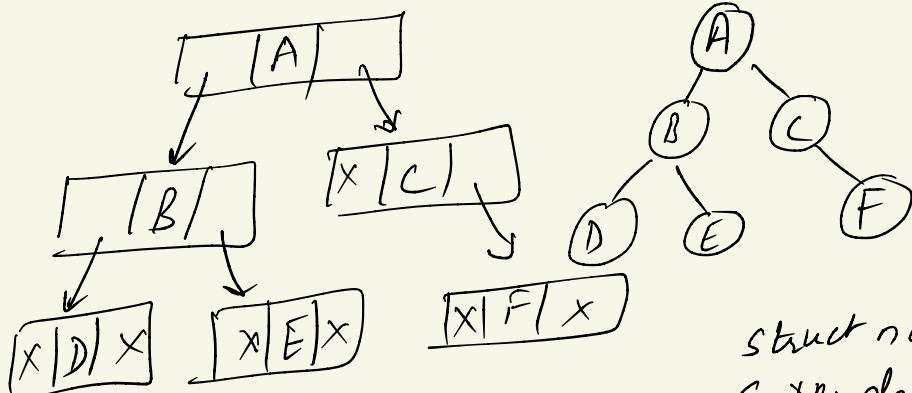
$$\text{degree}(M) = 0$$

degree of tree = max degree amongst all the nodes

height of tree = Total no. of levels

height of node = no of edges in the longest path from that node to a leaf node

$$\text{height}(B) = 2$$



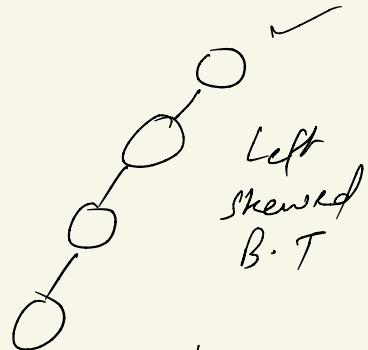
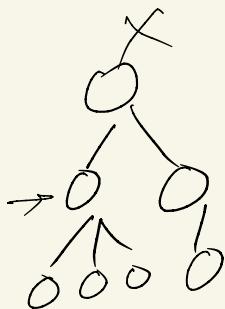
Advantages

- File system
- Routing protocols
- Binary search tree

```
struct node  
{  
    char data;  
    struct node *left;  
    struct node *right;  
};
```

Binary tree \Rightarrow each node at most has

2 children



Left
skewed
B.T

max nodes at any level $i \Rightarrow 2^i$

$n =$ Max no of nodes for any tree
of height $h \Rightarrow$

$$2^0 + 2^1 + 2^2 + \dots + 2^h$$

$\underbrace{\qquad\qquad\qquad}_{G.P \text{ series}}$

$$= 2^{h+1} - 1$$

Ques Assignment Min no of nodes for a tree
of height $h \Rightarrow h+1$

$$n = 2^{h+1} - 1$$

$$n+1 = 2^{h+1}$$

$$\log_2(n+1) = \log_2 2^{h+1}$$

$$\lceil \log_2 n + 1 - 1 \rceil = \min \text{ height}$$

$$n = h+1$$

$$h = (n) - 1$$

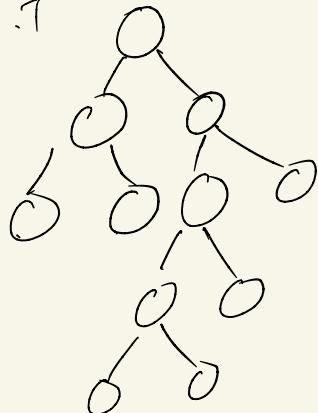
Max
height

Binary tree types

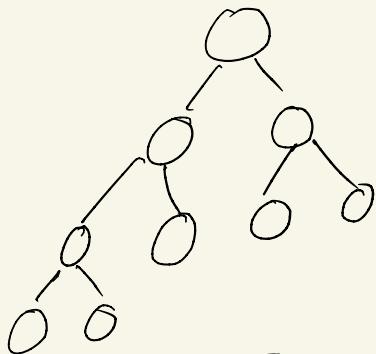
- Full / proper / strict \Rightarrow every node can have either 0 or 2 children
- Complete binary tree \Rightarrow child node assignment would be from left to right
- Perfect binary tree
- Degenerate binary tree

✓ F. B.T

~~X~~ CBT

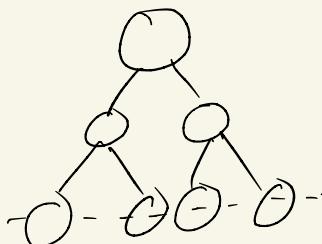


C. B.T

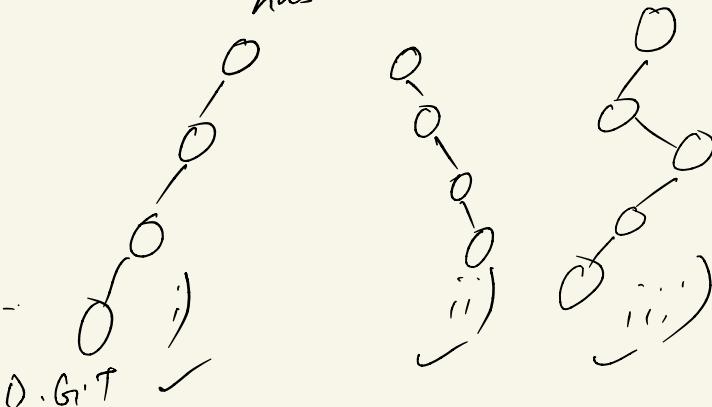


Perfect B.T

all the leaf
nodes are at
same level



Degenerate B.T
every internal node
has 1 child



	Max nodes	Min. nodes
B.T	$2^{h+1} - 1$	
F.B.T	$2^{h+1} - 1$	
C.B.T	$2^{h+1} - 1$	
F.B.T		$\frac{2^{h+1} - 1}{2^h}$ <p style="text-align: right;">Ques Assignment</p>

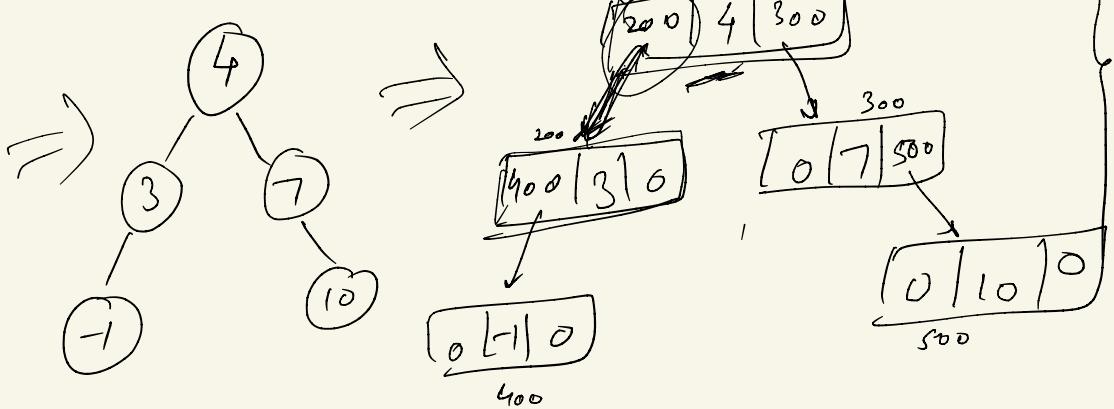
$$0 \Rightarrow 1$$

$$1 \Rightarrow 3$$

$$2 \Rightarrow 5$$

$$3 \Rightarrow 7$$

Binary tree implementation



struct node

```
|| { int data;
  || struct node *left;
  || struct node *right;
  || }
```

struct node* create() \Rightarrow recursively

while ($\{ \text{int } n; \text{printf}("Do you want to create nodes: "); \}$

$\text{scanf}("%d", &\text{newnode} \rightarrow \text{data});$ newnode = (struct node^*) malloc (size of (struct node));

$\text{printf}("Enter data:");$

$\text{scanf}("%d", &\text{newnode} \rightarrow \text{data});$ $= n$

$\text{newnode} \rightarrow \text{left} = \text{create}();$

$\text{newnode} \rightarrow \text{right} = \text{create}();$

void main()

```
{ struct node *root;
root = 0;
root = create(); }
```

