Data Structures & Algorithms -I

Lecture on 01-04-2020

Dr. Prerana Mukherjee

Introduction on Trees

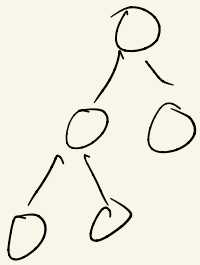
Data Structures

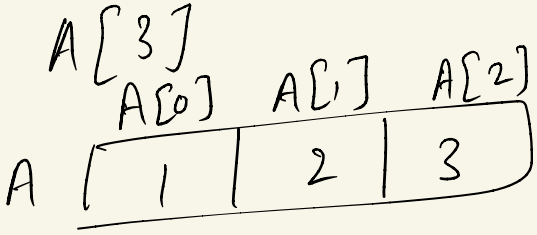
Linear
- one after other
- Sequential access
- order

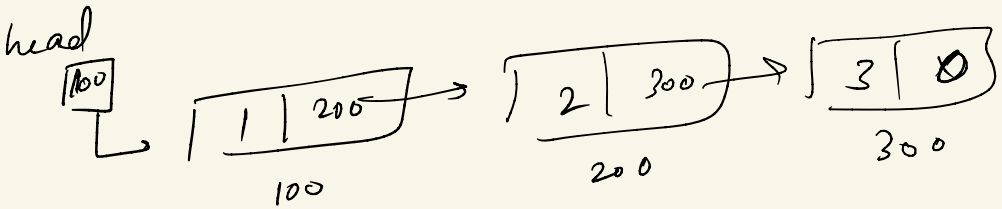
{ Arrays, Linked List, Stacks, Queues }

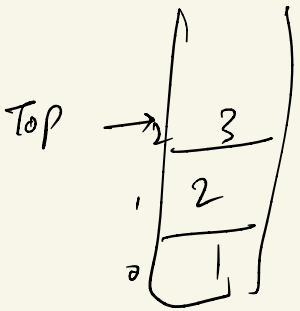
Non-linear
- hierarchy
- multiple levels

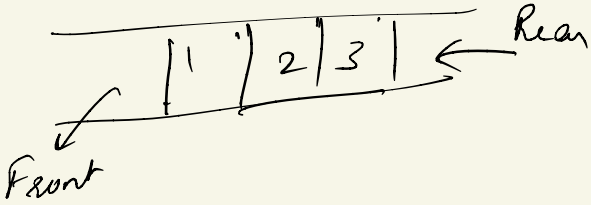
{ Trees, Graphs }

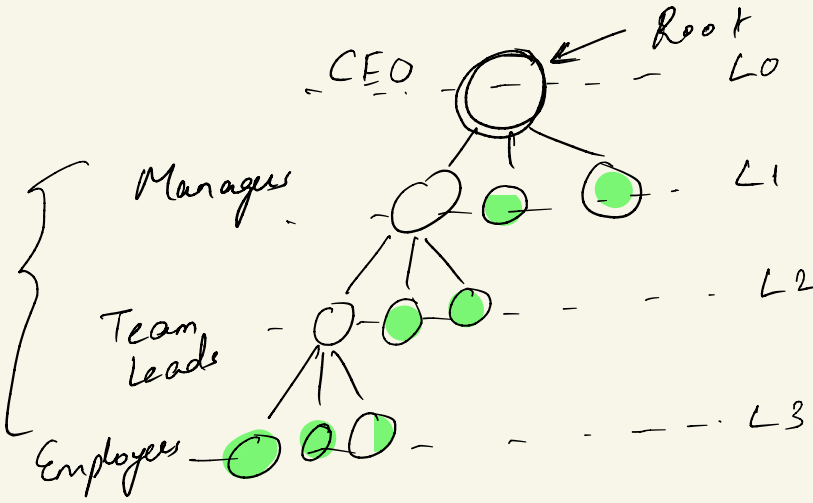


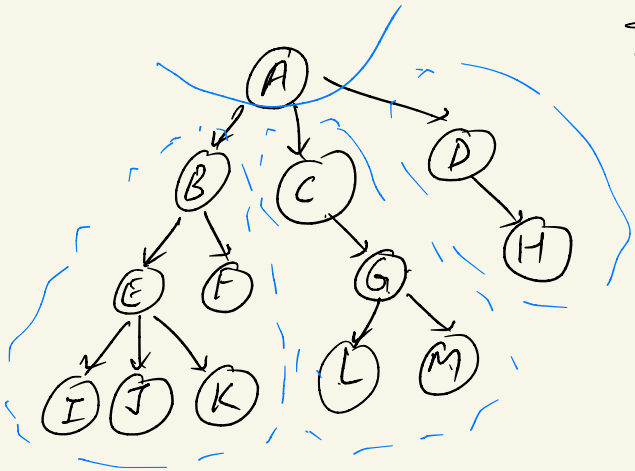












Tree can be defined as a collection of entities (nodes) to simulate some form of hierarchy

subtree

Root = A

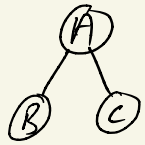
External / Leaf nodes = I, J, K, F, L, M, H

Internal / Non-leaf

Intermediate = B, C, D, E, G

Parent
ancestor

child
descendent



degree of node = no of children at that node

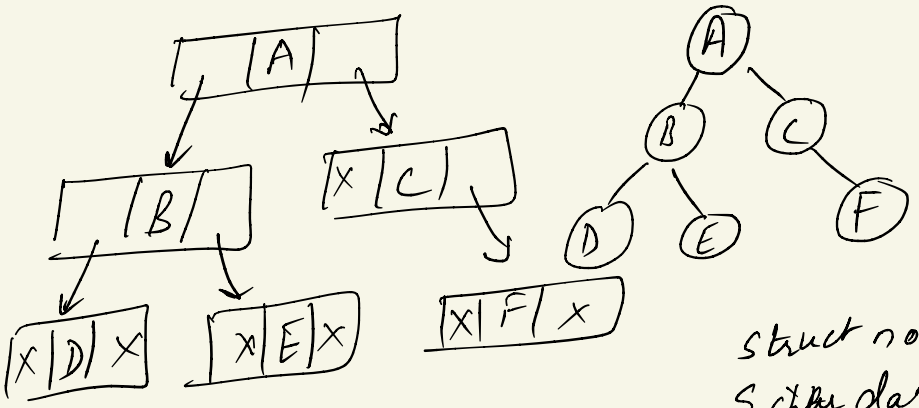
degree (A) = 0

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

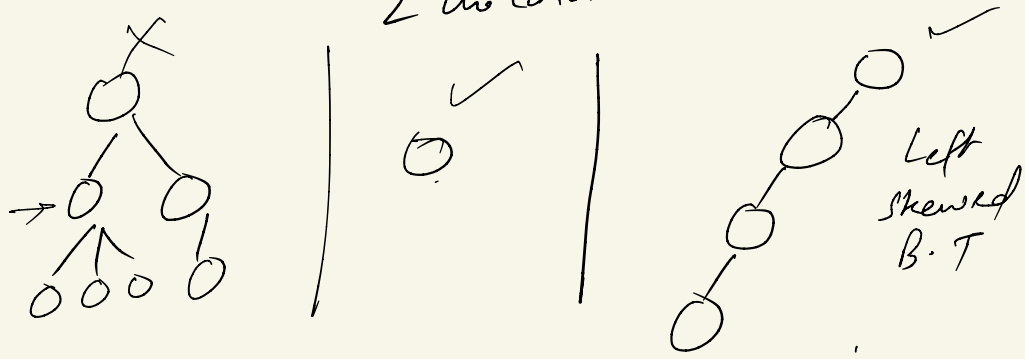
height (B) = 2



Advantages
→ File system
→ Routing protocols
→ Binary search tree

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

Binary tree ⇒ each node at most has 2 children



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

n = Max no of nodes for any tree of height h ⇒

$$\underbrace{2^0 + 2^1 + 2^2 + \ldots + 2^h}_{\text{G.P series}}$$

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

Ques Assignment Min no of nodes for a tree of height h ⇒ h+1

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

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

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

$$\lceil \log_2 n+1 - 1 \rceil = h$$

Min height

$$n = h+1$$

$$\boxed{h = (n) - 1}$$

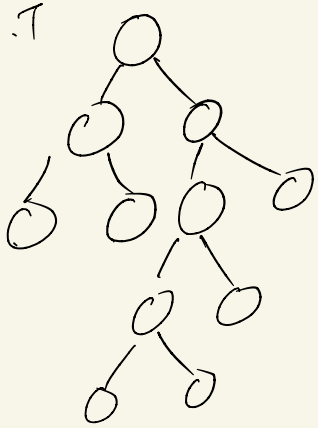
Max height

Binary tree types

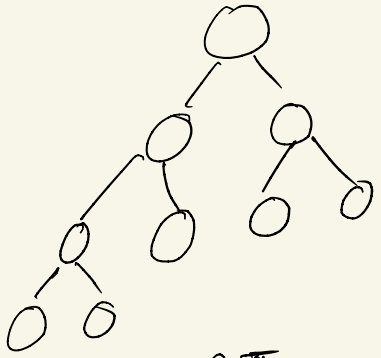
- Full / proper / strict ⇒ every node can have either 0 or 2 children
- Complete binary tree ⇒ child node assignment would be from left to right
- Perfect binary tree
- Degenerate binary tree

✓ F.B.T

✗ CBT

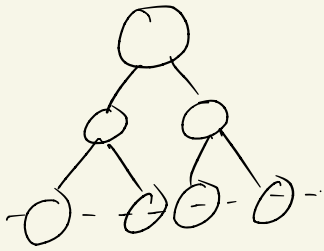


C.B.T



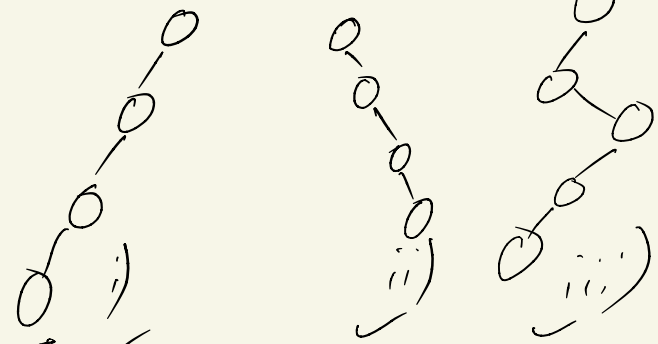
Perfect B.T

all the leaf nodes are at same level



Degenerate B.T

every internal node has 1 child



D.G.T ✓

	Max nodes	Min. nodes
B.T	$2^{h+1}-1$	$h+1$
F.B.T	$2^{h+1}-1$	$2h+1$
C.B.T	$2^{h+1}-1$	2^h

Ques Assignment

F.B.T

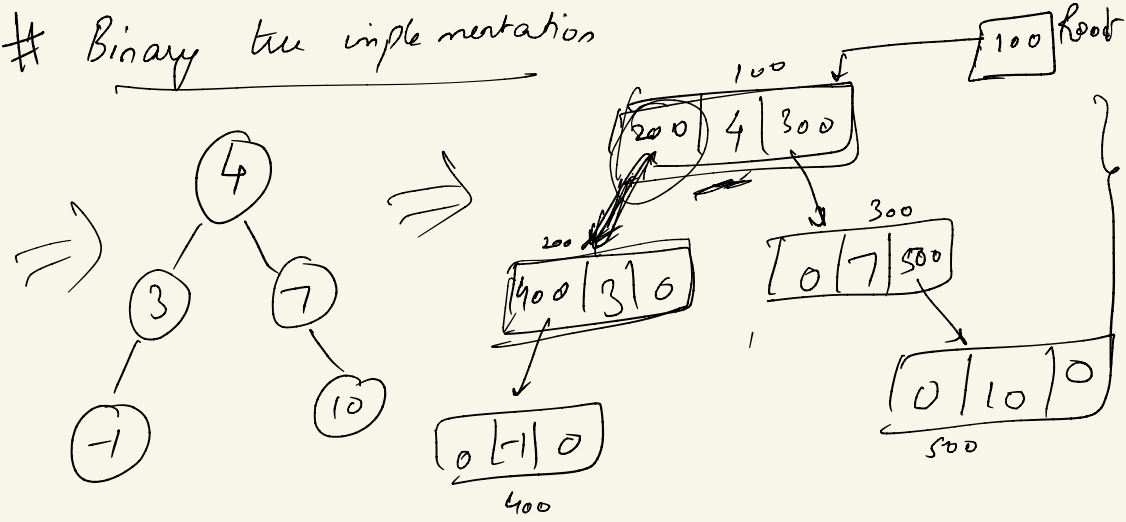
$0 \Rightarrow 1$

$1 \Rightarrow 3$

$2 \Rightarrow 5$

$3 \Rightarrow 7$

Binary tree implementation




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

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

```
struct node * create()
{ int x;
  printf("Do you want to create nodes:");
  scanf("%d", &x);
  while(x)
  { struct node *newnode;
    newnode = (struct node *) malloc (sizeof
                                      (struct node));
    printf("Enter data:");
    scanf("%d", &newnode->data);
    newnode->left = create();
    newnode->right = create();
```