

Tree Basics

- Tree intro
- Tree terminology
- Tree traversal
- Basic Problems

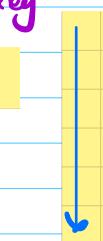
Linear

data
structures

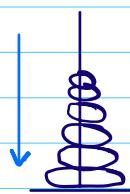
Array

3 5 -2 4 99 ...

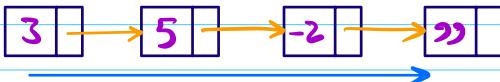
key Hashset



stack

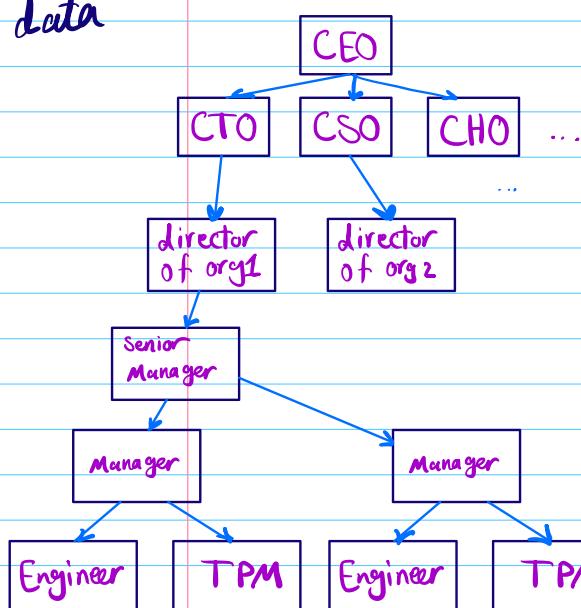


link list



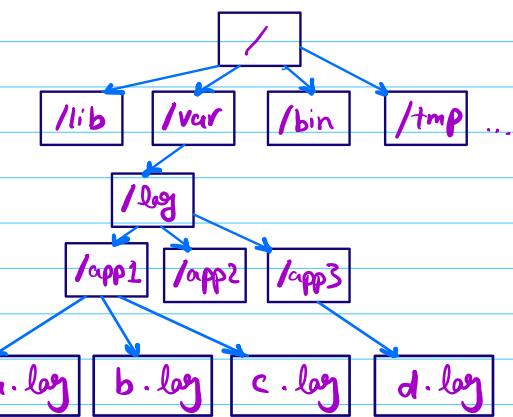
Hierarchical
data

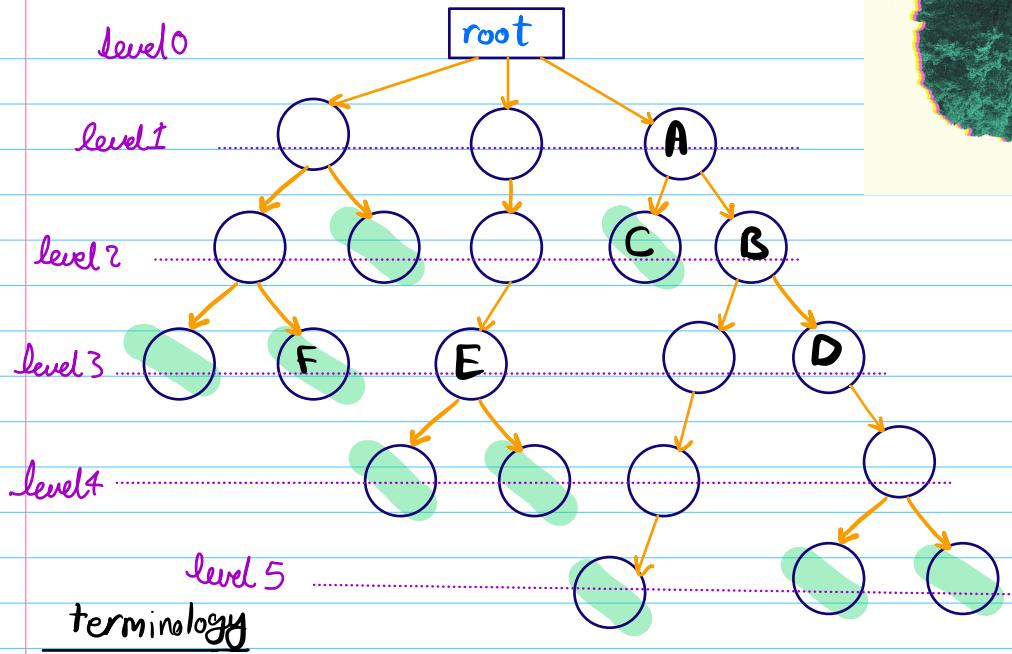
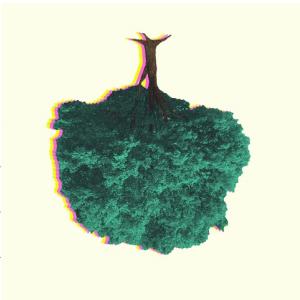
ex: Company Org



ex: Family tree

ex: File & Dir on
Computer (linux)





terminology

A is **parent** of B
B is **child** of A
C & B are **sibling**

F, E & D same level

root: the node with no parent

leaf: the node with no child

root not parent

Quiz

Quiz

Tree:

- only one root
- each node has only one parent

Height

length of longest path

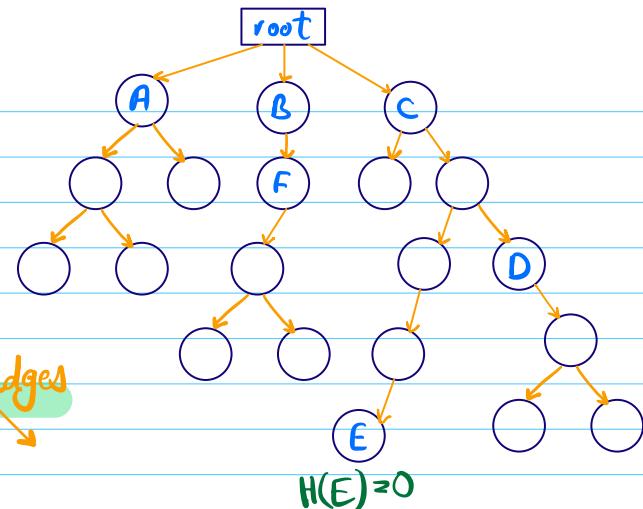
from one node to any
of descendant leaf
node

path is measured \rightarrow # of edges

$$H(A) = 2$$

$$H(B) = 3$$

Quiz $H(C) - H(E) = 4 - 0 = 4$



$$H(E) = 0$$

depth

length of path from root

$$D(A) = 1$$

$$D(F) = 2$$

Quiz

$$D(E) = 5$$

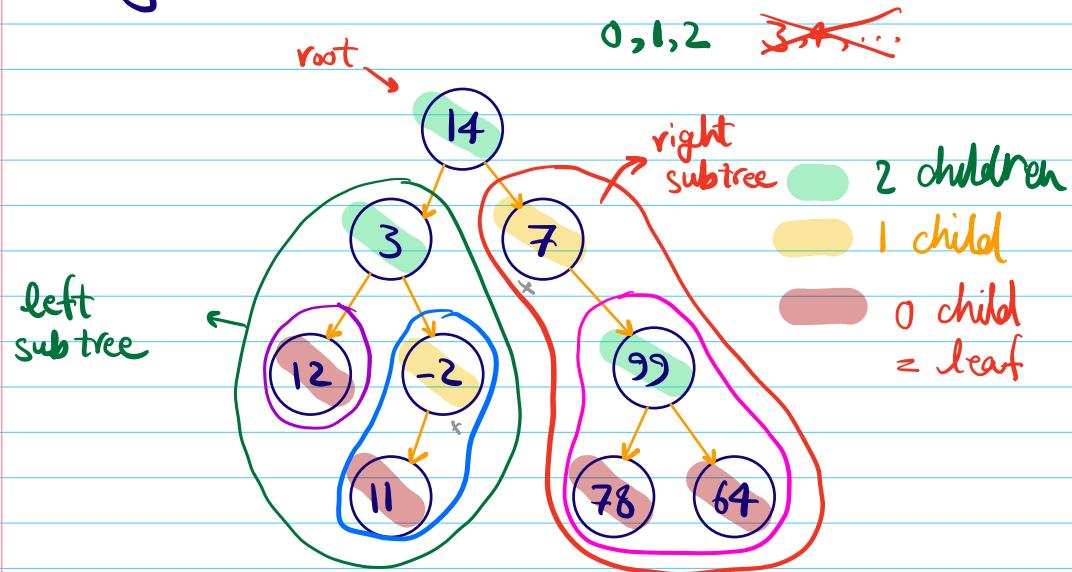
$$D(D) = 3$$

important's height & depth def. in your assignments
& HW might be slightly different

Height of tree height of root

Depth of tree = or
 max depth of any leaf

Binary Trees all nodes have max 2 children



class Tnode{

int val; //v

Tnode left; //l

Tnode right; //r

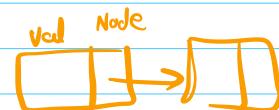
public Tnode(int v){

this.val = v;

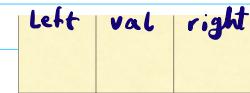
left = null;

right = null;

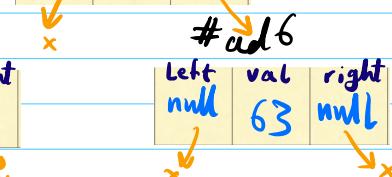
}



Tnode



r.right = new Tnode(32)



r.left = new Tnode(21)



tree
class construction → Advanced

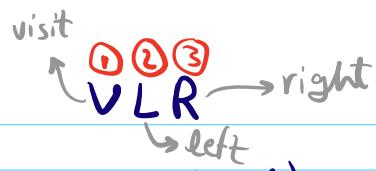
Tree Traversal

today { -preorder | -level order
 | -inorder | -vertical level order
 | -post order advanced
 classy

break?

Subtree

Preorder

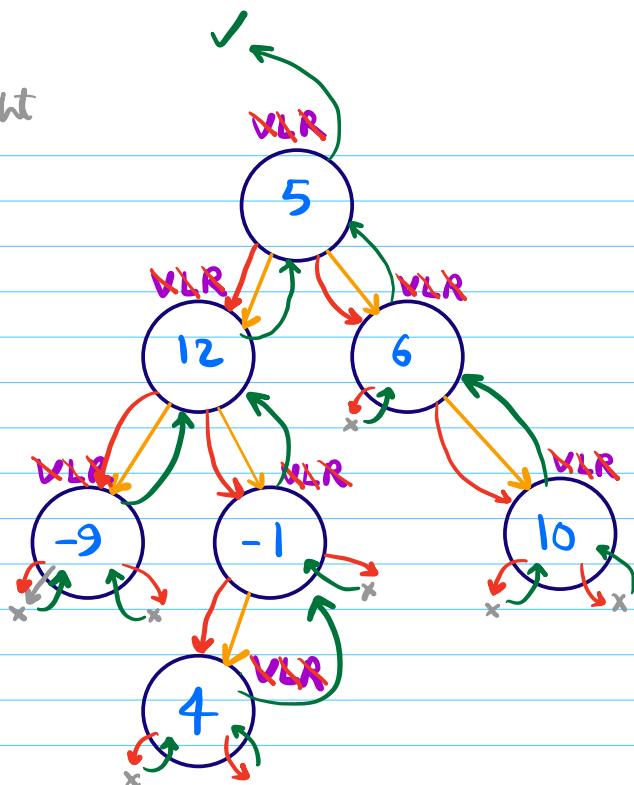


step1: print (root.val)

step2: go to left subtree

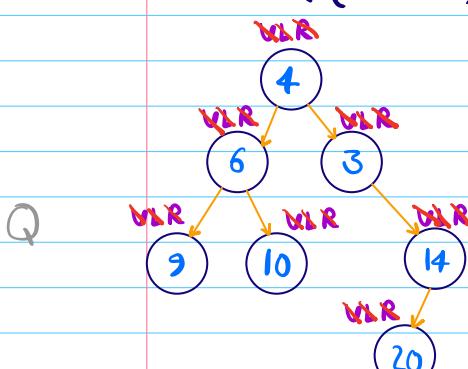
and do preorder traversal

step3: go to right subtree and do preorder traversal



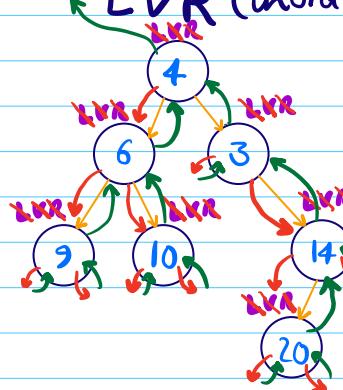
output: 5, 12, -9, -1, 4, 6, 10

VLR (preorder)



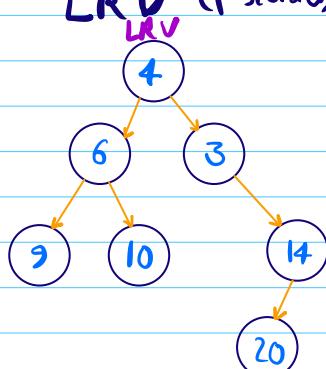
ans = 4, 6, 9, 10, 3, 14, 20

LVR (inorder)



9, 6, 10, 4, 3, 20, 14

LRV (postorder)



9, 10, 6, 20, 14, 3, 4

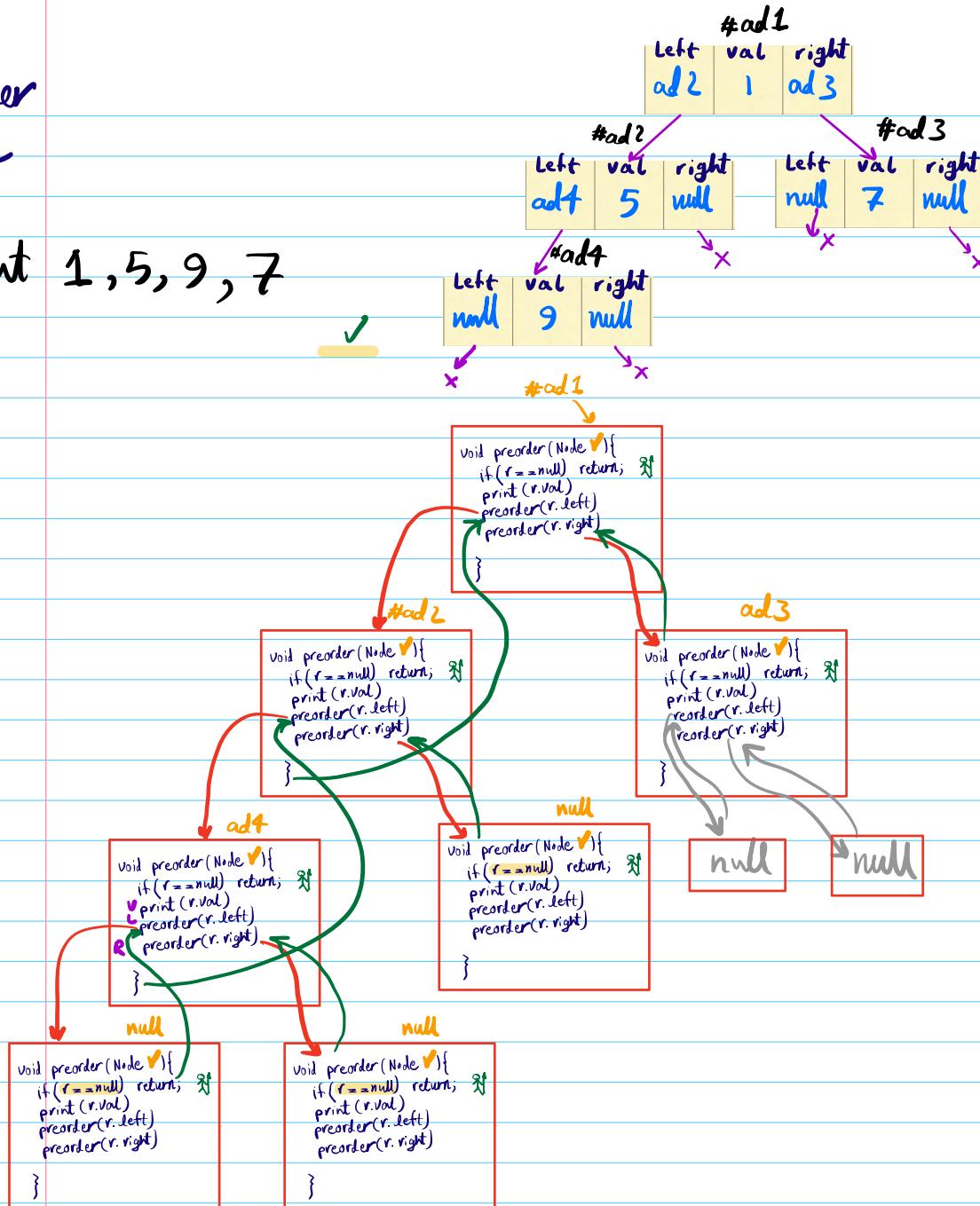
void preorder(Node r) {
 Q & Q if ($r == \text{null}$) return; 
 TC: $O(n)$ print(r.val)
 SC: $O(H)$ preorder(r.left)
 height preorder(r.right)
 trace? }

LVR void inorder(Node r) {
 Inorder if ($r == \text{null}$) return; 
 TC: inorder(r.left)
 SC: print(r.val)
 inorder(r.right)
 }

LRV void postorder(Node r) {
 postorder if ($r == \text{null}$) return;
 TC: postorder(r.left)
 SC: postorder(r.right)
 print(r.val)
 }

preorder
trace

output 1, 5, 9, 7



Three

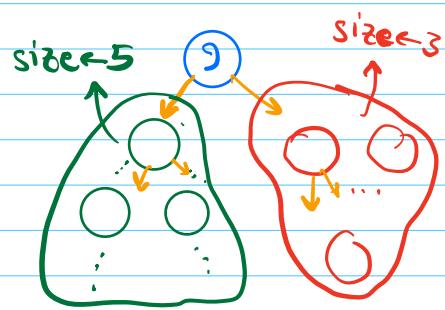
Tree problems // All 3 problem with recursion,
no global variable

① size(Node r), Number of nodes

② sum(Node r), sum of all nodes values

③ Height(Node r) Assignment

$5+1+3$

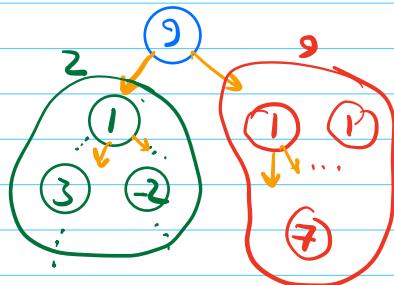


ans for ① int size (Node r){
 if ($r == \text{null}$) return 0;
 l s = size(r.left)
 // dummy visit
 r s = size(r.right)
 ret l + 1 + r;

ans for ② int sum(Node r){

$2+9+9 = 20$

 if ($r == \text{null}$) return 0;
 l s = sum(r.left)
 // dummy visit
 r s = sum(r.right)
 ret l + r.v + r;



doubt
Session

Josephus problem recursion-28

$A = 100 \leftarrow$

$B = 5 \leftarrow$

$$A - 1 = 99$$

int J (int A , int B) {
if ($A == 1$) ret 1

int $S = J(A-1, B);$

ret $[S + (B-1)] \% (A+1)$

