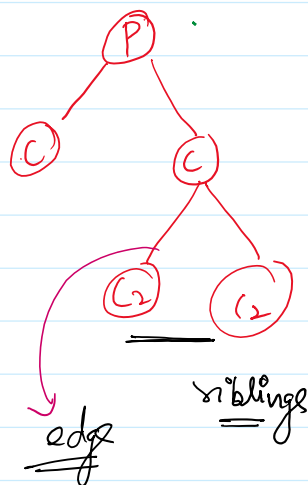


Trees

→ hierarchical

→ what is hierarchy



first node is root

Balanced

unbalanced

read definitions

Level

height

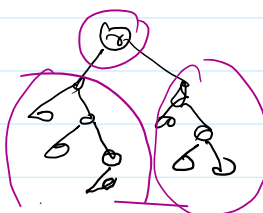
LST

RST

skewed trees

left sub tree

LST



RST

memory

Notes

Notes

1

Notes

(Repo)

memory

Trees

Arrays
Linked List

build on top of it

array
ll

or array + ll

Stack & Queue

lesson plan

→

Binary Tree

0, 1, 2

0 - 2

children

h

0, 1, 2

2

at max 2

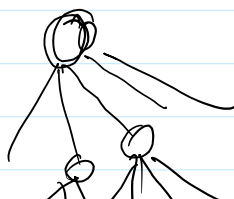


Generic Trees

Tree

0 - n

1 children



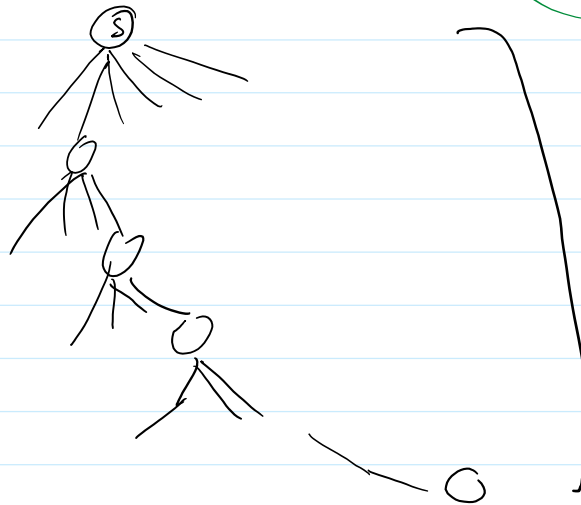


N-ary Tree

ternary

Quad Trees

0 - 4

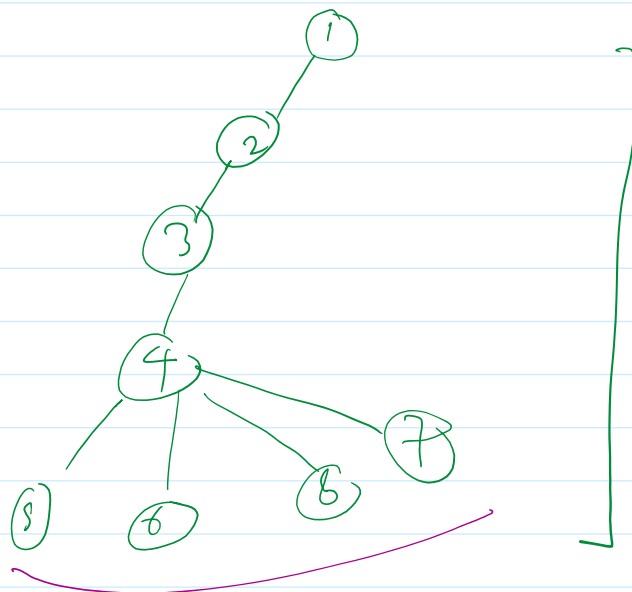


array $[(p, c), (p, c), \dots]$

(p, c)

(p, c) mid
op1

$(-1, 1)$ $[(1, 2), (2, 3), (3, 4), (4, 5), (4, 6), (4, 8), (4, 7)]$



How will you implement it

Node

class Node {

int data;

Node prev, next;

}

2

ref

n

~~ArrayList<Node> children;~~

→ implement

linked list → working

arr (p, c)

[level order traversal]

create if

print if

→ we will see

4

at

max

two

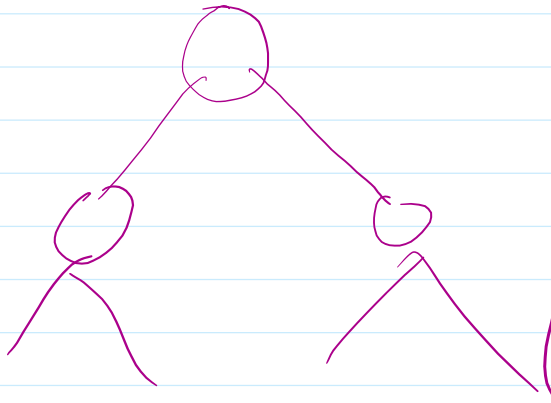
children

Binary Tree

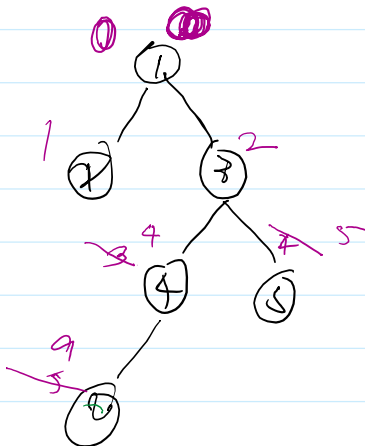
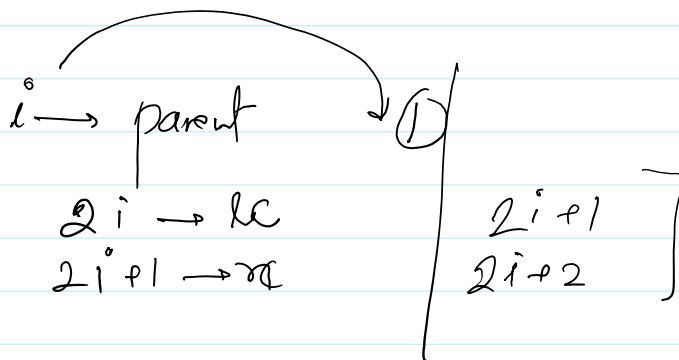
→ Binary tree

Class in
left, right

Binary Tree



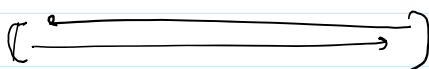
Arrays

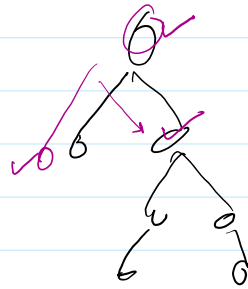
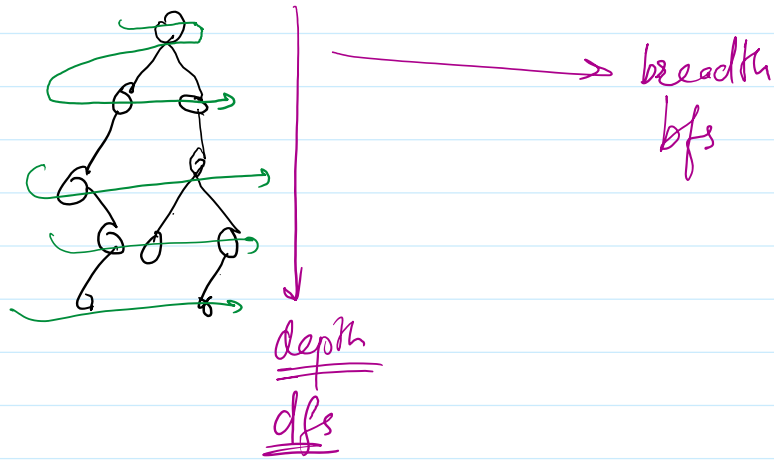


$\{ \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{matrix} \}$
 $\{ \begin{matrix} 1 & 7 & 3 & & 4 & 5 & & & & 9 & \end{matrix} \}$

implement Both

Traversal





LST

RST

↑ L ↑ R ↑ $\emptyset \rightarrow \text{root}$

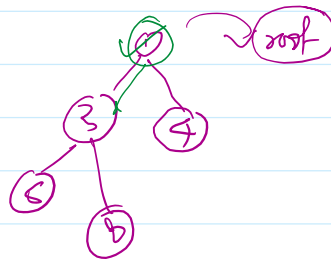
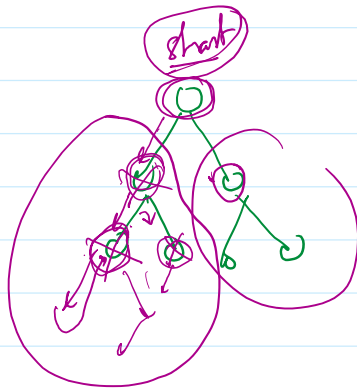
RLR | LrR | LRL

first

2

process

recursive



RLR

printing

$f(\text{Node root})$

$\{$ root data $\rightarrow n$

$f(\text{root.left})$

$f(\text{root.right})$

$\emptyset \rightarrow \text{root}$