

STACK

TOP only

[7, 2, 0
2, 7, 0

* Peek \rightarrow TOP's val

* POP \rightarrow REMOVE TOP + RET
PUSH \rightarrow ADD NEW TOP

isEmpty \rightarrow Bool

STACK AS LL



"VOLATILE"

"DESTRUCTIVE" \leftarrow RAUFER SAZ

$O(N)$

TOP TO BOTTOM

CHECK NEXT ??

WHILE (STACK.POP())

≡

3

* RECURSIVELY

QUEUE

FIFO

LINE AT STORE

* enqueue

* dequeue

peek

isEmpty



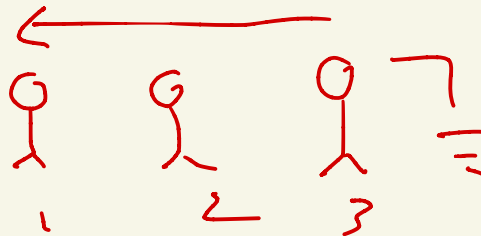
TRAVERSAL

while (peek) {

== dequeue

}

"VOLATILE"

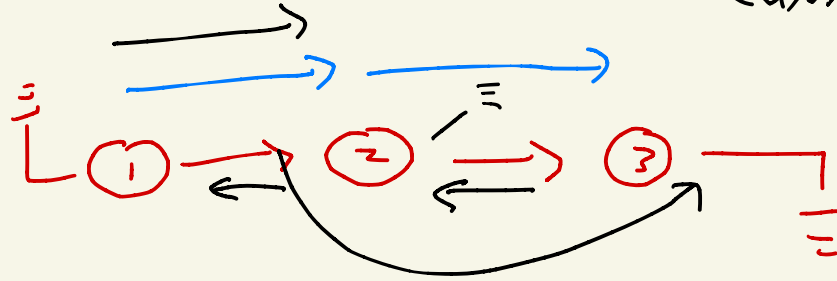


LINKED LIST

{ SLOW/FAST
TURTLE/MARE
LEADER/FOLLOW
RUNNER

NODES

- VALUE
- NEXT



HEAD - 1ST

TAIL - LAST
NULL

TRAVERSAL

WHILE (CURRENT.NEXT) {

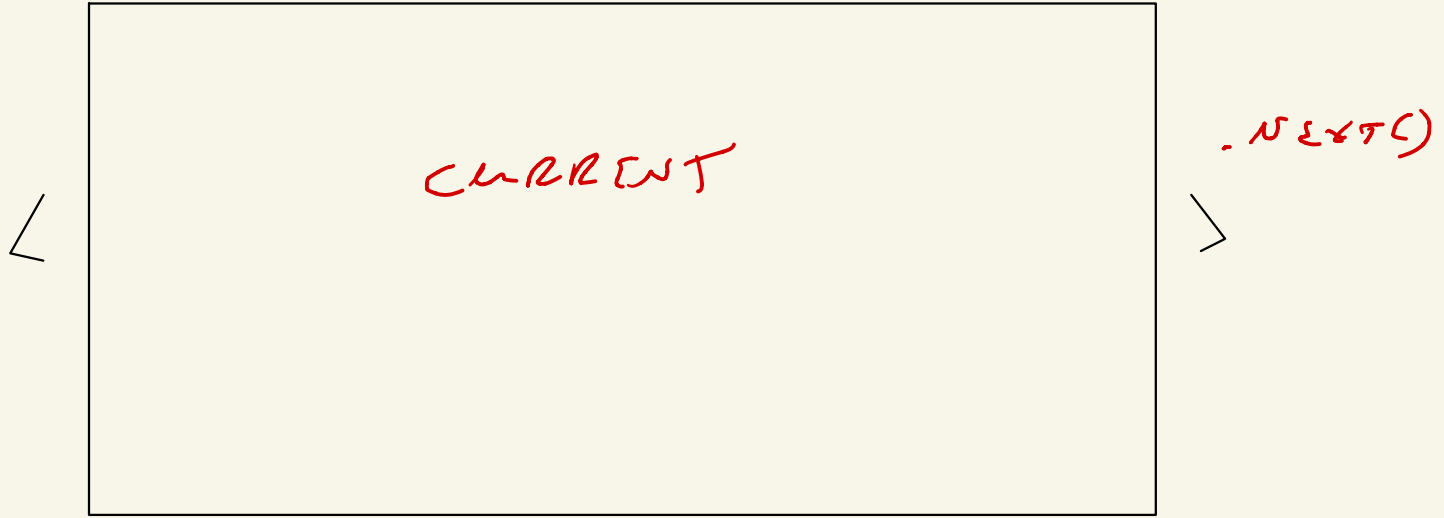
HEAD ↓

CURRENT.NEXT.NEXT

}



SLIDER



FOLLOWERS
1ST

TREES

HIERARCHY
: ORG CHART
: HTML

ROOT

NODE
: VALUE
: LEFT
: RIGHT

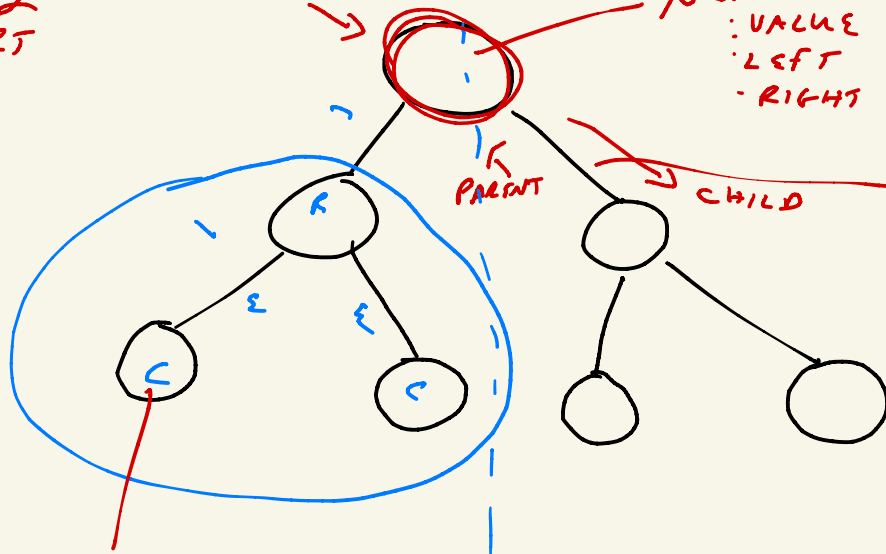
PARENT

CHILD

EDGE

$H = 4$

$H = 2$



LEAF
NODE

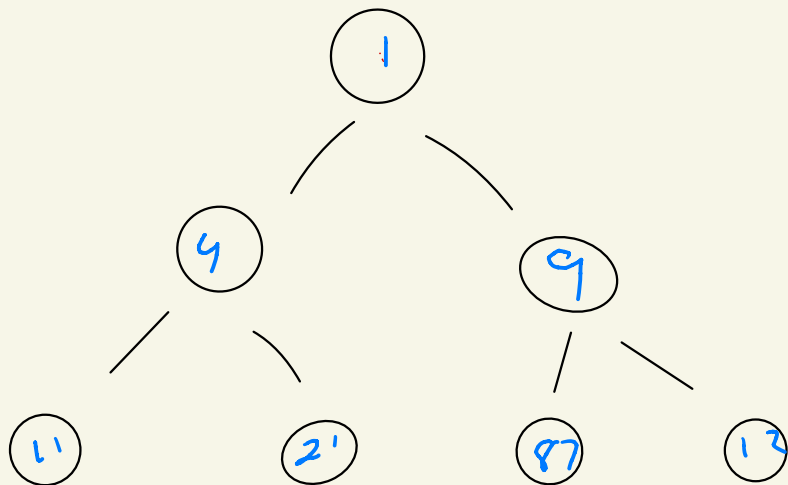
MAX. HEIGHT =

EDGES
FROM
ROOT
TO LEAF

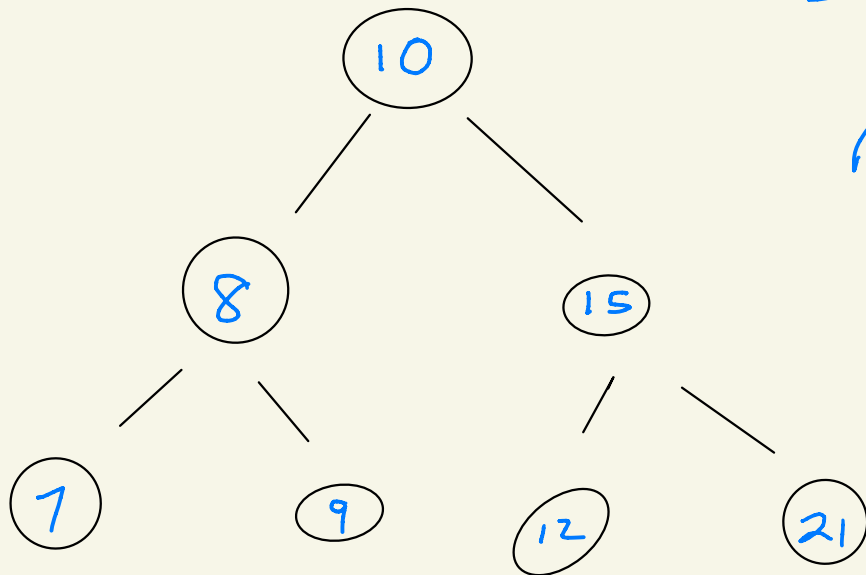
WIDTH = # LEAVES

BINARY TREE

ONLY:
LEFT
RIGHT



BINARY SEARCH TREE



LEFT = LESS
THAN

RIGHT = GREATER
THAN

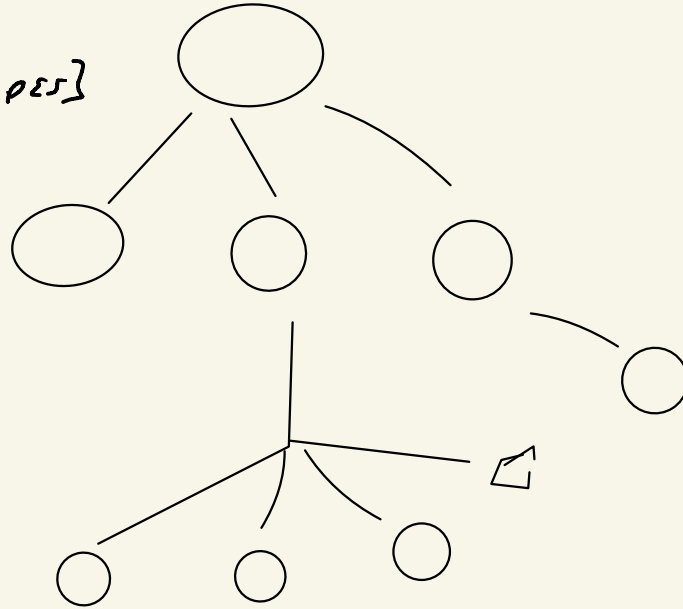
. ADD

K-ARY TREE

NODE

VALUE

CHILDREN; [nodes]




BIG O (RECURSIVE)

↑
SPACE

STACK - N

Q - N

LL - N

TREE - 

$O(h+n)$