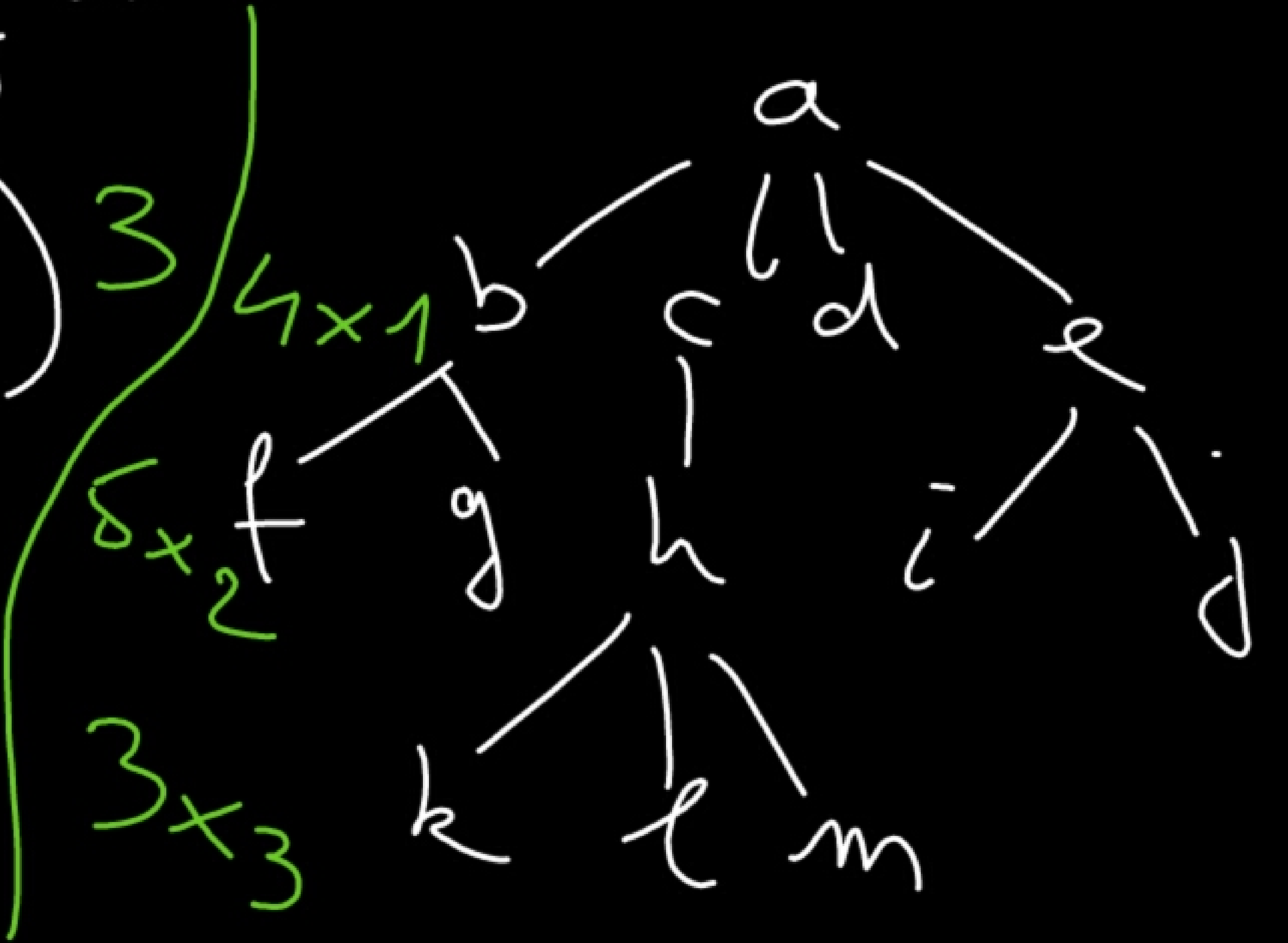
Definitions

- size : number of elements   13
- height (depth) of an element (node) : number of links between root and elt
- height (depth) of a tree : $\max\limits_{e \in t} (\text{height}(e))$ 3
- pathlength of a tree : $\sum\limits_{e \in t} (\text{height}(e))$ 23



ADT extension Tree, Forest

Operations

size : Tree $\to$ Integer

sizef : Forest $\to$ Integer

height : Tree $\to$ Integer

heightf : Forest $\to$ Integer

PL : Tree $\to$ Integer

PLF : Forest $\to$ Integer

Axioms

A_1 $size(new(e,f)) \equiv 1 + sizeF(f)$

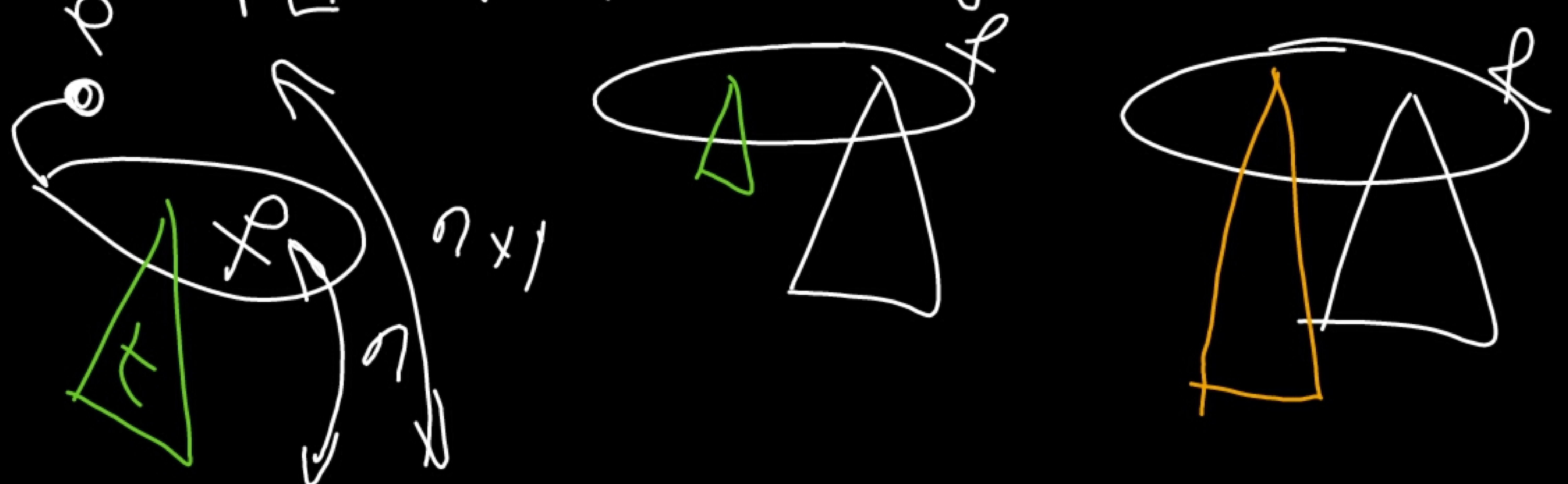
A_2 $sizeF(newf) \equiv 0$

A_3 $sizeF(addTree(f,t,n)) \equiv sizeF(f) + size(t)$

A_4 $hight(new(e,f)) \equiv heightf(f)+1$

A_5 $heightF(newf) \equiv -1$

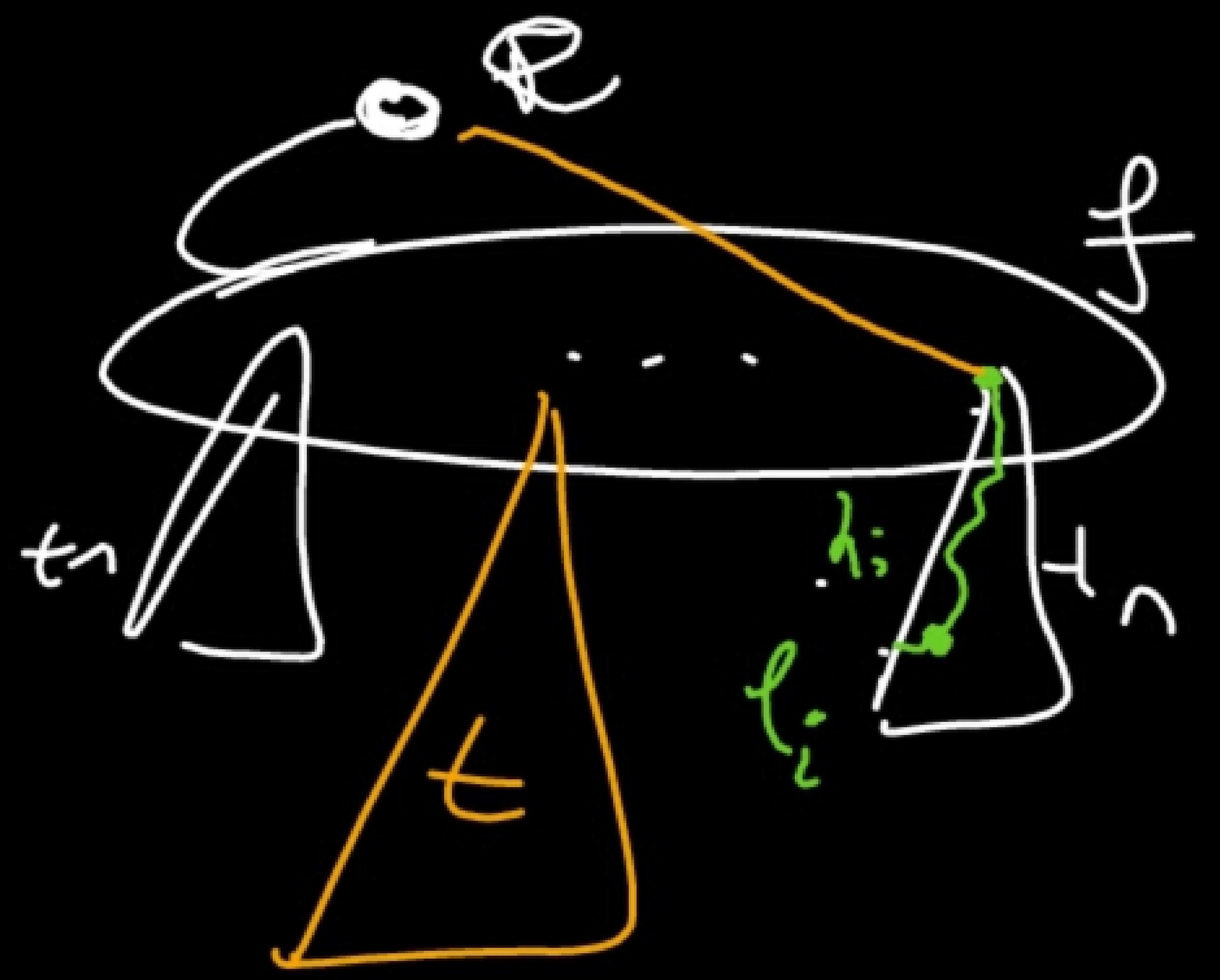
A_6 $heightF(addTree(f,t,n)) \equiv max(height(t), heightF(f))$



$A_7 \quad PL(\mathrm{new}(e, f)) \equiv PLF(f) + \mathrm{sizeF}(f)$

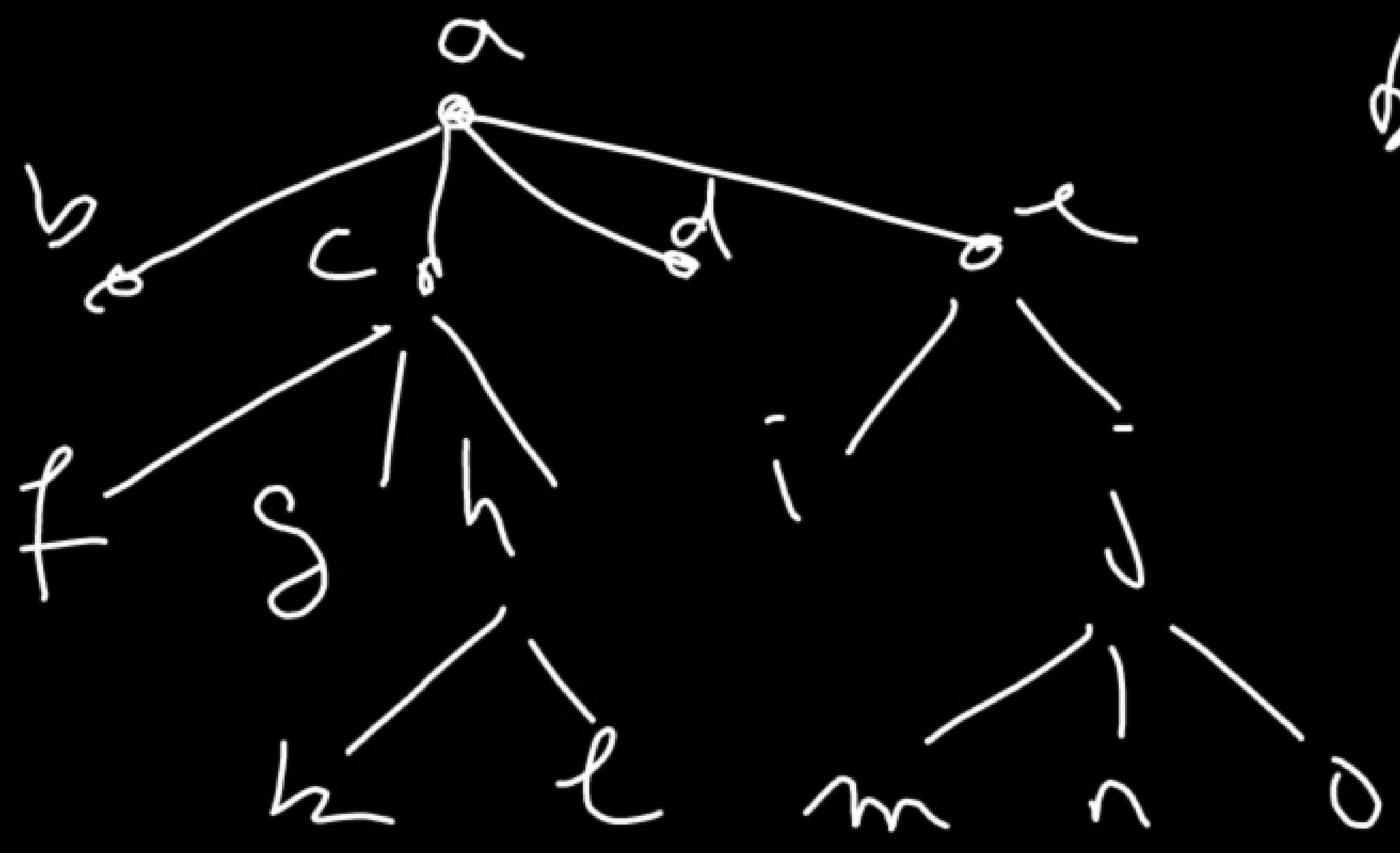
$A_8 \quad PLF(\mathrm{newF}) \equiv 0$

$A_9 \quad PLF(\mathrm{addTree}(f, t, n)) \equiv PLF(f) + PL(t)$



Enumerating elements of a tree! Many ways to do that

* breadth-first : list all elements at depth ℓ before listing elements at depth $\ell+1$
* depth-first : list all elements in subtree n before listing elements in subtree $n+1$



breadth-first, left-right traversal:

a b c d e ... o

depth-first, 1 → n:

a b c f g h k l d e i j m n o

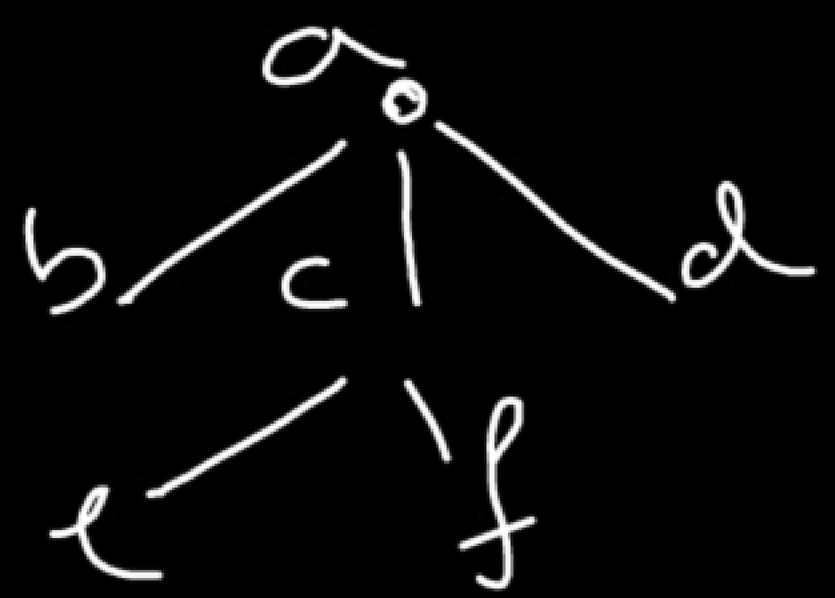
Binary trees (nodes have <u>at most</u> 2 subtrees)

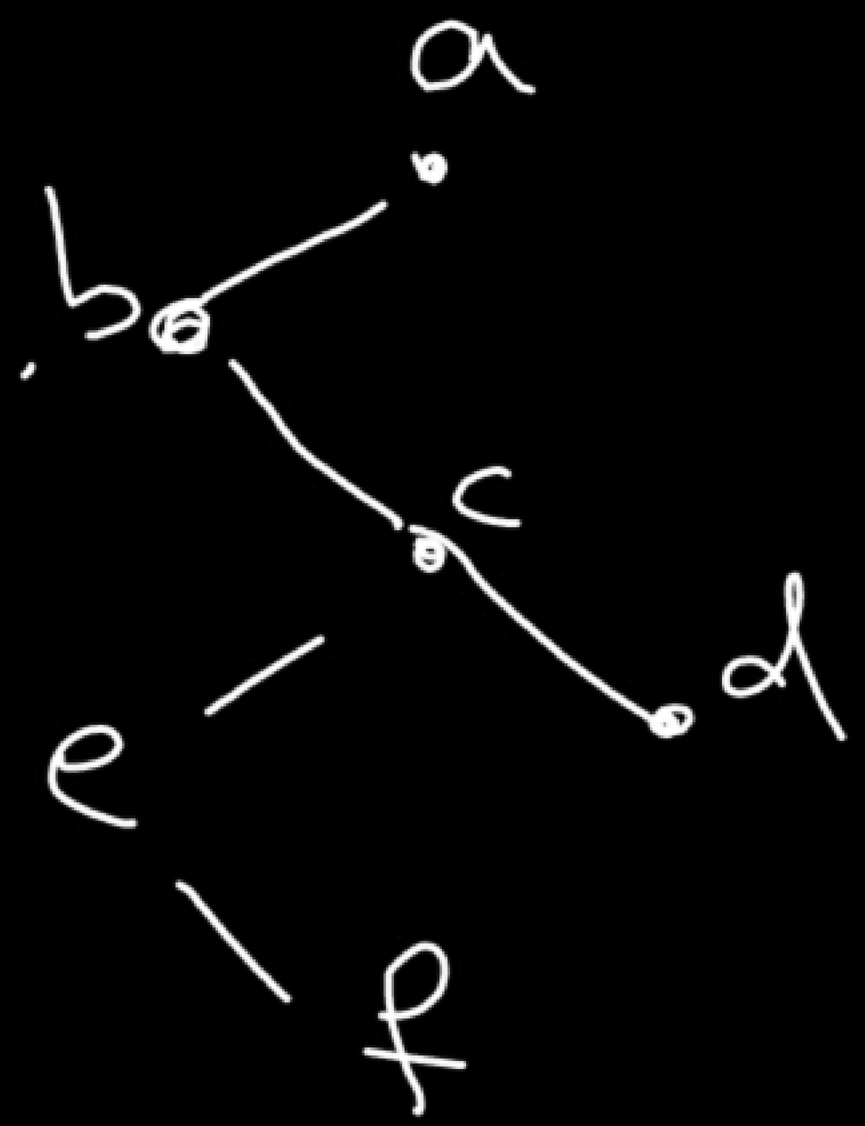
1) easier to deal with than general trees

2) represent the same information without loss of performance

Transformation:

- root → root
- 1st subtree → left child
- right sibling → right child


a
b
c
d
e
f


a
b
c
d
e
f

Binary Trees (BT)

a BT is
* either empty
* or made of
  1) a root element
  2) a left child (BT)
  3) a right child (BT)

ADT BT

Use Element, Boolean

Operations

* new : $\longrightarrow$ BT
* $\langle \_, \_, \_ \rangle$: Element $\times$ BT $\times$ BT $\longrightarrow$ BT

root : BT $\longrightarrow$ Element

left : BT $\longrightarrow$ BT

right : BT $\longrightarrow$ BT

contains : BT $\times$ Element $\longrightarrow$ Boolean

preconditions

root (t), iff t $\not\equiv$ new

left ______

right ______

Axioms

A_1 $root(\langle r, L, R\rangle) \equiv r$

A_2 $left(\langle r, L, R\rangle) \equiv L$

A_3 $right(\langle r, L, R\rangle) \equiv R$

A_4 $contains(new, e) \equiv F$

A_5 $contains(\langle r, L, R\rangle, r) \equiv T$

A_6 $contains(\langle r, L, R\rangle, e) \equiv contains(L, e) \text{ or } contains(R, e)$

$\longrightarrow$ extension with size, height and PL?

ADT extension BT

Use Integer

Operations

size : BT $\longrightarrow$ Integer

height : BT $\longrightarrow$ Integer

PL : BT $\longrightarrow$ Integer

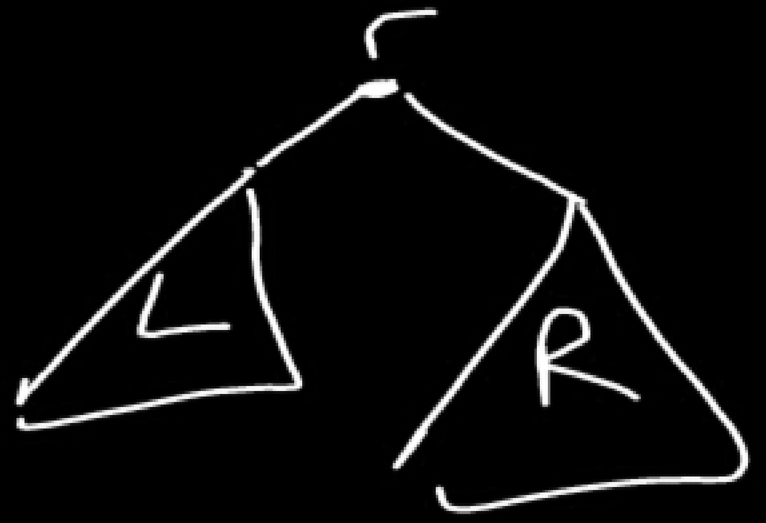
Axioms

A_1 size(new) $\equiv$ 0

A_2 size(<r, L, R>) $\equiv$ size(L) + size(R) + 1

A_3 height(new) $\equiv$ -1

A_4 height(<r, L, R>) $\equiv$ max(height(L), height(R)) + 1



A_5 PL(new) $\equiv$ 0

A_6 PL(<r, L, R>) $\equiv$ PL(L) + PL(R) + size(L) + size(R)

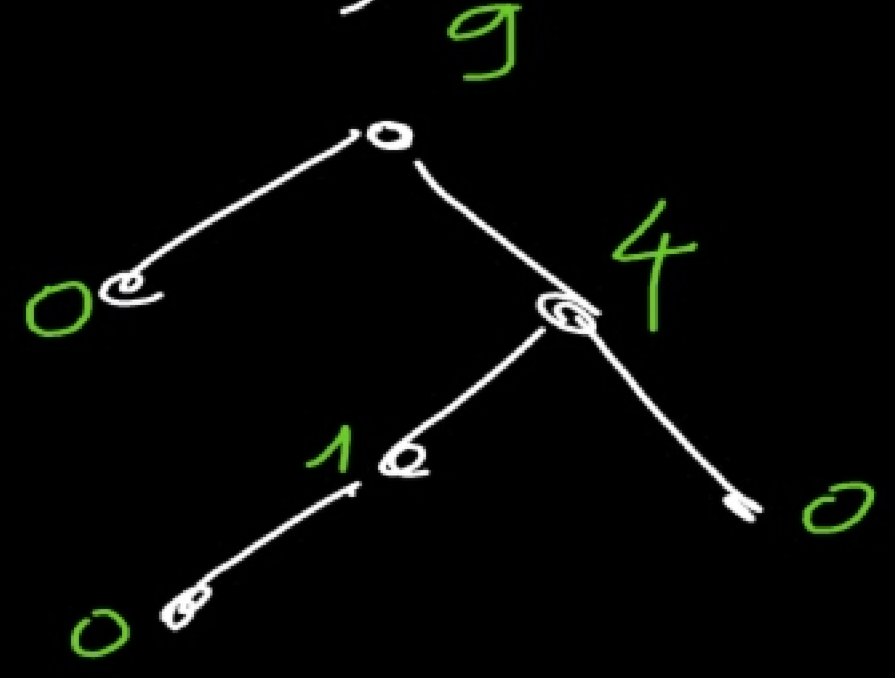
1×0

2×1

2×2

1×3

9



Depth-first traversal:

each node is visited 3 times (visit1, visit2, visit3)

```
depthfirst(t) {
    if (t != new) {
        visit1(root(t))
        depthfirst(left(t))
        visit2(root(t))
        depthfirst(right(t))
        visit3(root(t))
    }
}
```

preorder: visit2 = visit3 = ∅ A B D E G H C F

inorder: visit1 = visit3 = ∅ D B G E H A F C

postorder: visit1 = visit2 = ∅ D G H E B F C A

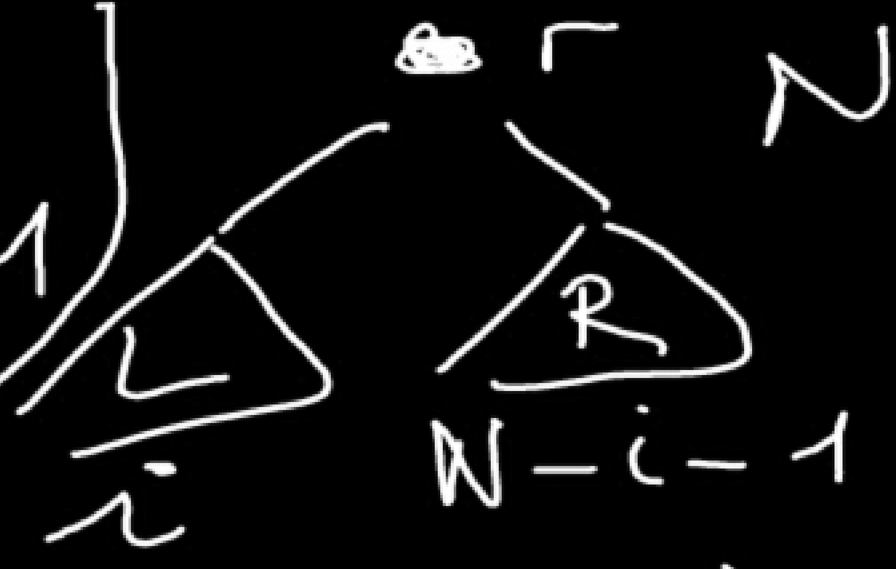
```
inorder(t){
    if (t != null) {
        inorder(left(t))
        visit(root(t))
        inorder(right(t))
    }
}
```

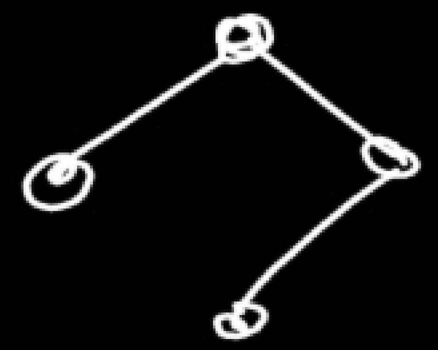
complexity?

unit op.: visit

param: N (size of t)

$$C(0) = 0$$
$$C(N) = C(i) + C(N-i-1) + 1$$





n	$C(n)$
0	0
1	1
2	2
3	3
4	4
n	n

hyp: $C(n) = n$?

prop: $C(n) = n \quad \forall n$

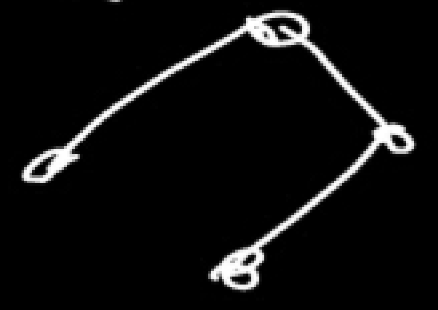
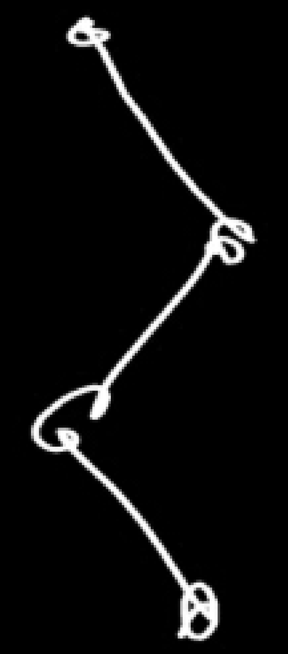
$p(0) = p(1) = p(2) = T$

hyp: $\forall i < n,\ p(i) = T$

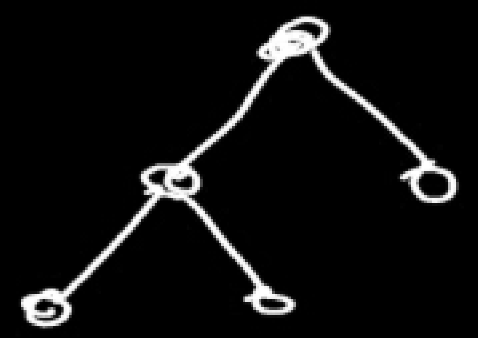
$$C(n) = C(i) + C(n-i-1) + 1$$
$$= \cancel{i} + n \cancel{-i} \cancel{-1} \cancel{+1} = n \quad \text{qed}$$

ex: t a BT of size N
min number of internal nodes?

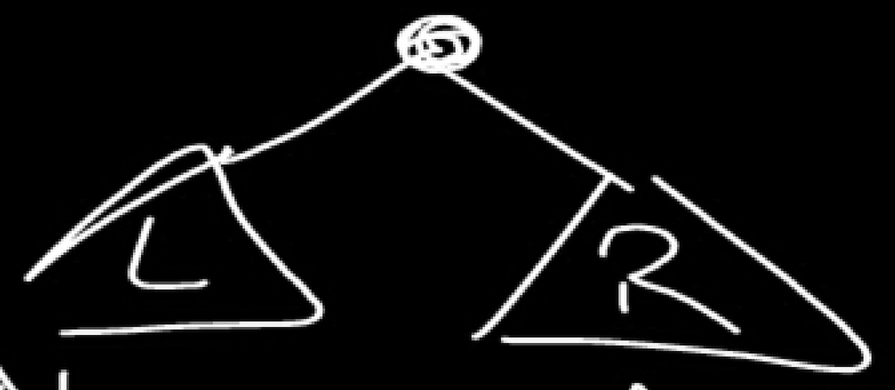
node = leaf or internal node

n	n_i
1	0
2	1
3	1
4	2
5	2
6	3



$$\frac{N-1}{2} \qquad \left\lfloor \frac{N}{2} \right\rfloor$$



N_L, N_{i_L} $\qquad$ N_R, N_{i_R}

prop(n): min internal nodes if size is n, $\frac{n-1}{2}$

$p(1) = T$

hyp: $\begin{cases} p(N_L) \\ p(N_R) \end{cases}$

$$N = N_L + N_R + 1$$

$$N_i = N_{i_L} + N_{i_R} + 1$$

$$N_i = \frac{N_L - 1}{2} + \frac{N_R - 1}{2} + 1$$

$$= \frac{N_L + N_R}{2} = \frac{N-1}{2} \quad \text{qed}$$

t a BT of size N.

number of links? $N-1$?

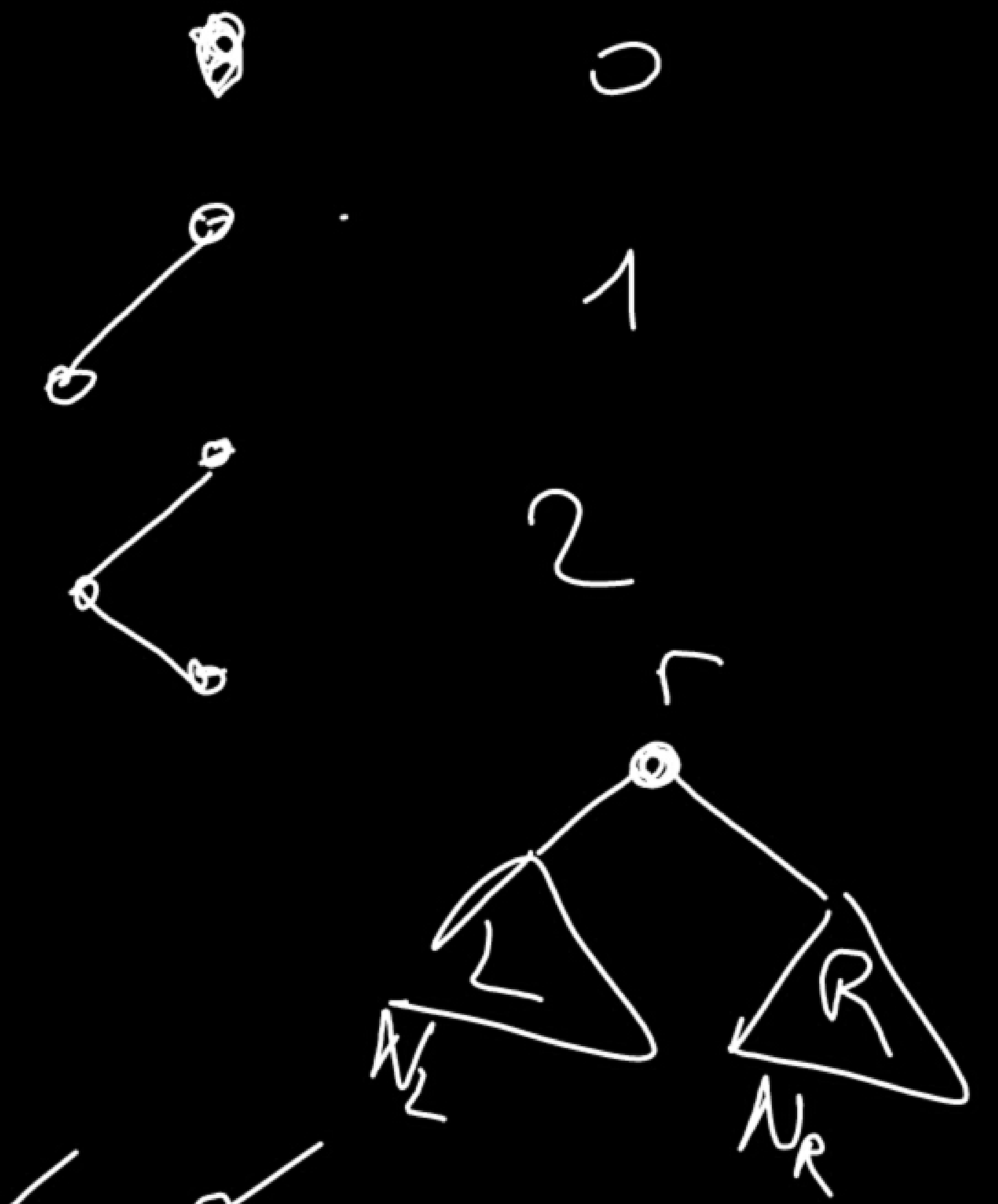
prop(n): if size is n, $n-1$ links

$p(1) = p(2) = T$

hyp: $\forall i < n,\ p(i) = T$

N: size

L: links



$$N = N_L + N_R + 1$$

$$L = L_L + L_R + 2$$

$$= N_L \cancel{-1} + N_R \cancel{-1} \cancel{+2}$$

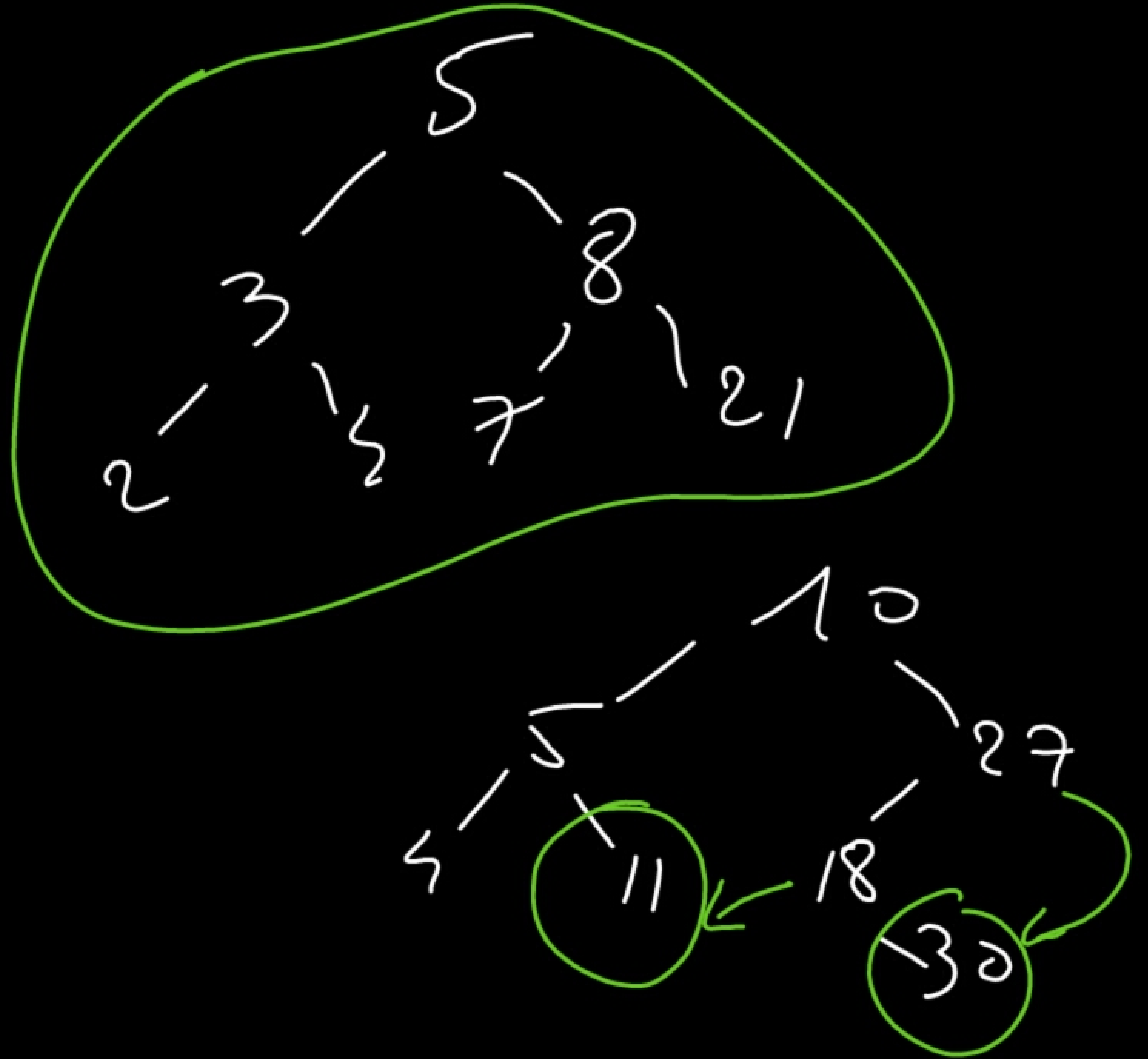
$$= N_L + N_R = N - 1 \quad \text{qed}$$

Binary search trees (BST)

<u>def</u>: t a BST is a BT /

$$\forall n_1, n_2 \in t^2$$

$$\text{if } n_1 < n_2 \rightarrow \begin{cases} n_1 \in left(n_2) \\ n_2 \in right(n_1) \end{cases}$$

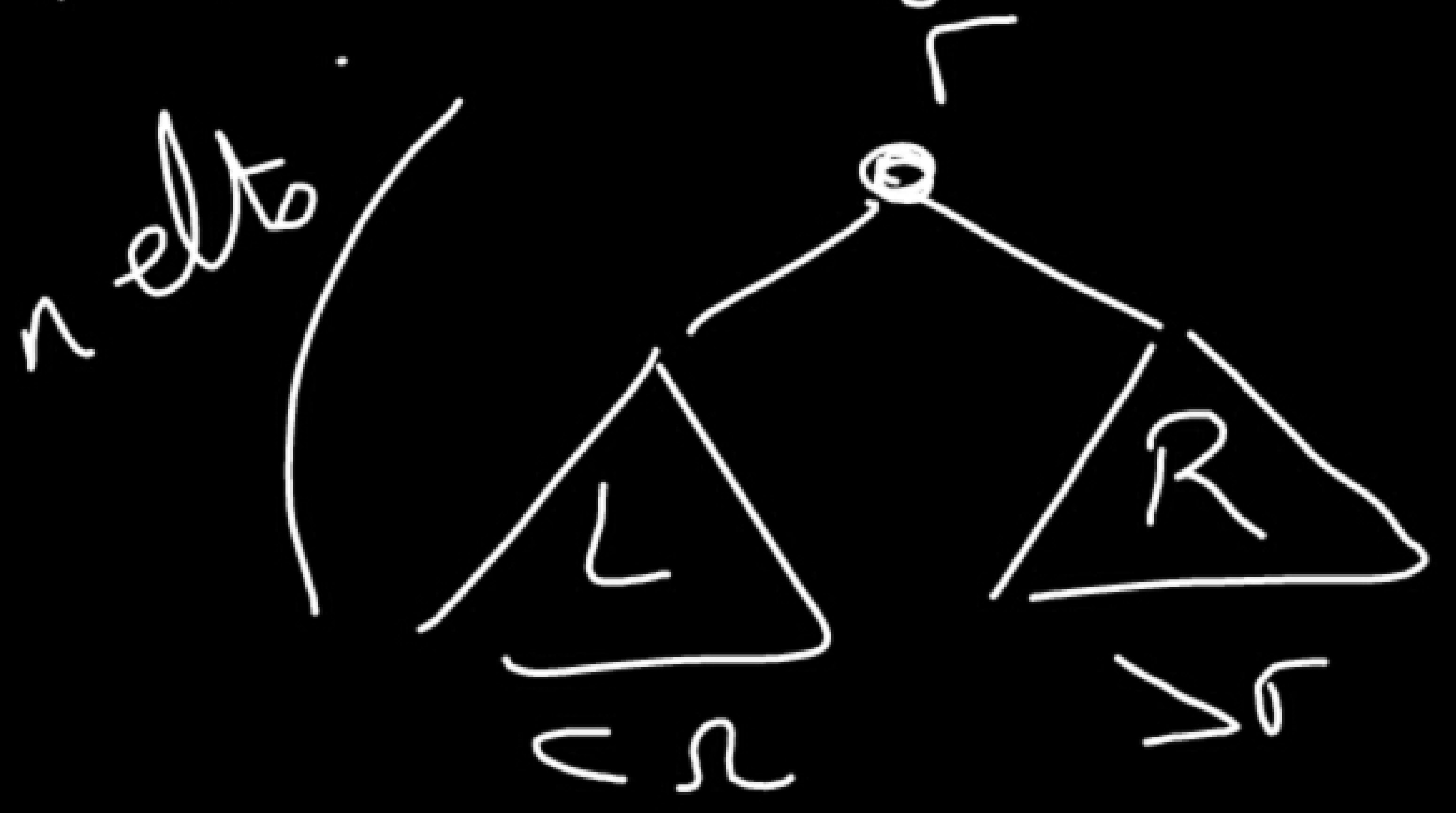


pre: 5 3 2 4 8 7 21

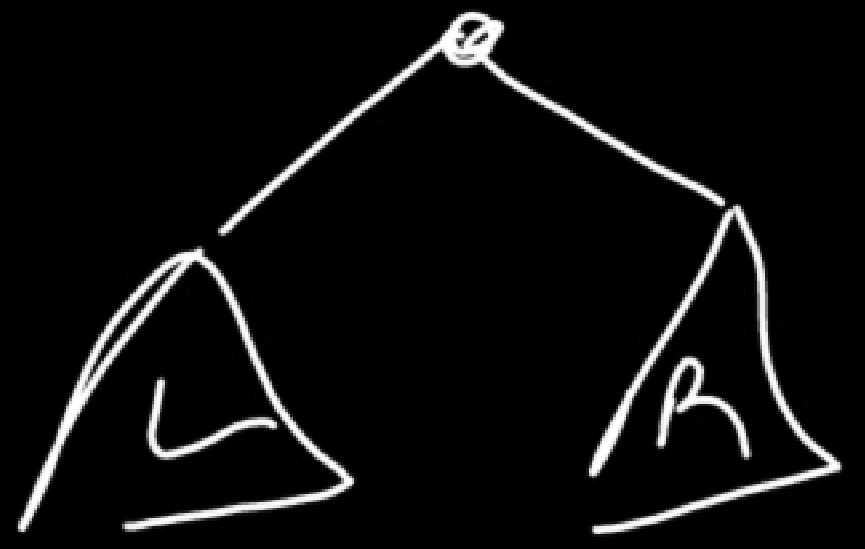
in: 2 3 4 5 7 8 21

Post: 2 4 3 7 21 8 5

prop: inorder gives a sorted list of elements in a BST



hyp $\forall i < n$, inorder(t) gives a sorted list



inorder(t):
inorder(L)
print(r)
inorder(R)

$\underbrace{e_1 \cdots\cdots e_i}_{\text{sorted}} \quad r \quad \underbrace{e'_1 \cdots\cdots e'_j}_{\text{sorted}}$

ADT BST extends BT

Use Element($\leq$)

Operations

contains : $BST \times Element \to Boolean$

add : $BST \times Element \to BST$

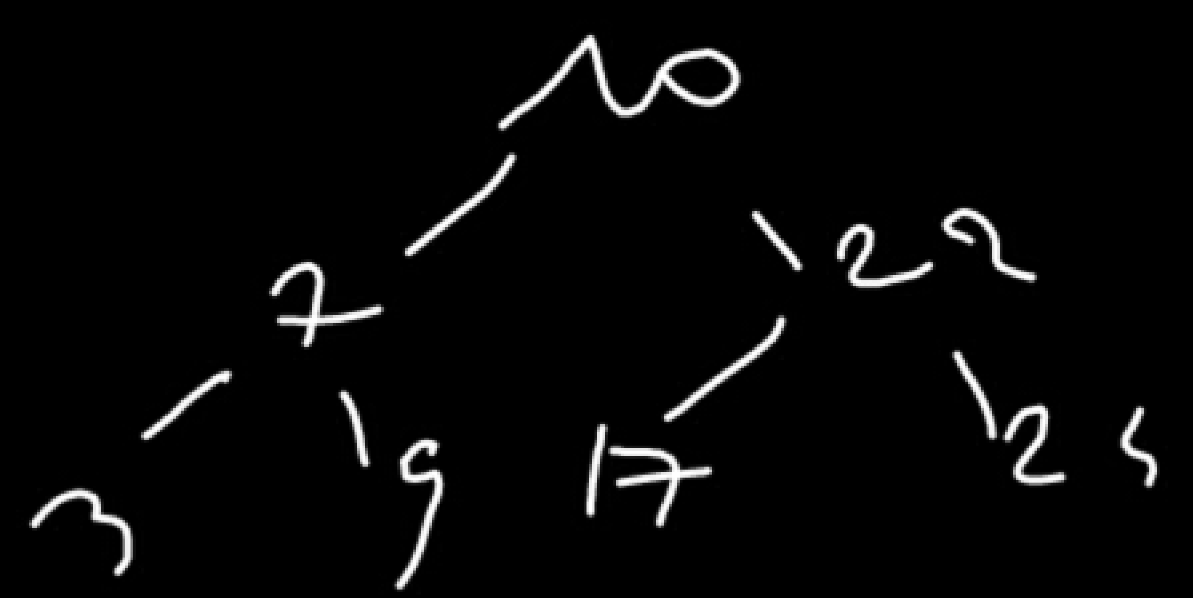
Axioms

A_1 $contains(new, e) \equiv F$

A_2 $contains(\langle r, L, R\rangle, r) \equiv T$

A_3 $e \leq r \Rightarrow contains(\langle r, L, R\rangle, e) \equiv contains(L, e)$

A_4 $contains(\langle r, L, R\rangle, e) \equiv contains(R, e)$



$contains(10, 22)$

A_4 $contains(22, 22)$

A_2 T

$contains(10, 21)$

A_4 $contains(22, 21)$

A_3 $contains(17, 21)$

A_4 $contains(new, 21)$

A_1 F

$A_5 \quad add(new, e) \equiv <e, new, new>$

$A_6 \quad add(<r, L, R>, r) \equiv <r, L, R>$

$A_6^* \quad add(t, root(t)) \equiv t$

$A_7 \quad e \leq r \Rightarrow add(<r, L, R>, e) \equiv <r, add(L, e), R>$

$A_8 \quad add(<r, L, R>, e) \equiv <r, L, add(R, e)>$