A_4 $\text{hight}(\text{new}(e, f)) \equiv \text{hightF}(f) + 1$

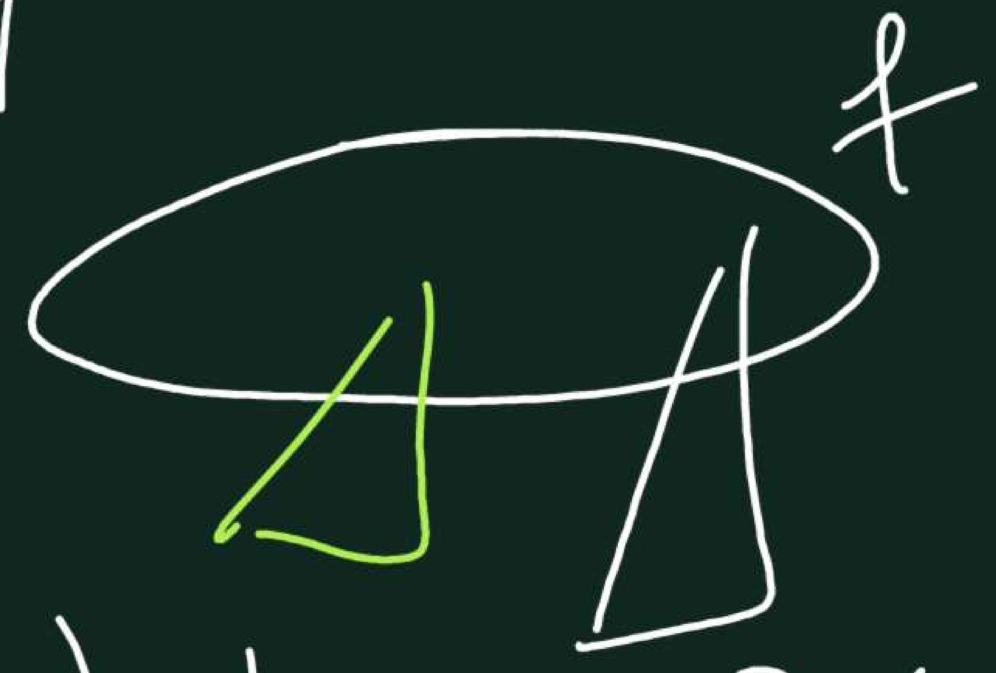
A_5 $\text{hightF}(\text{newf}) \equiv -1$

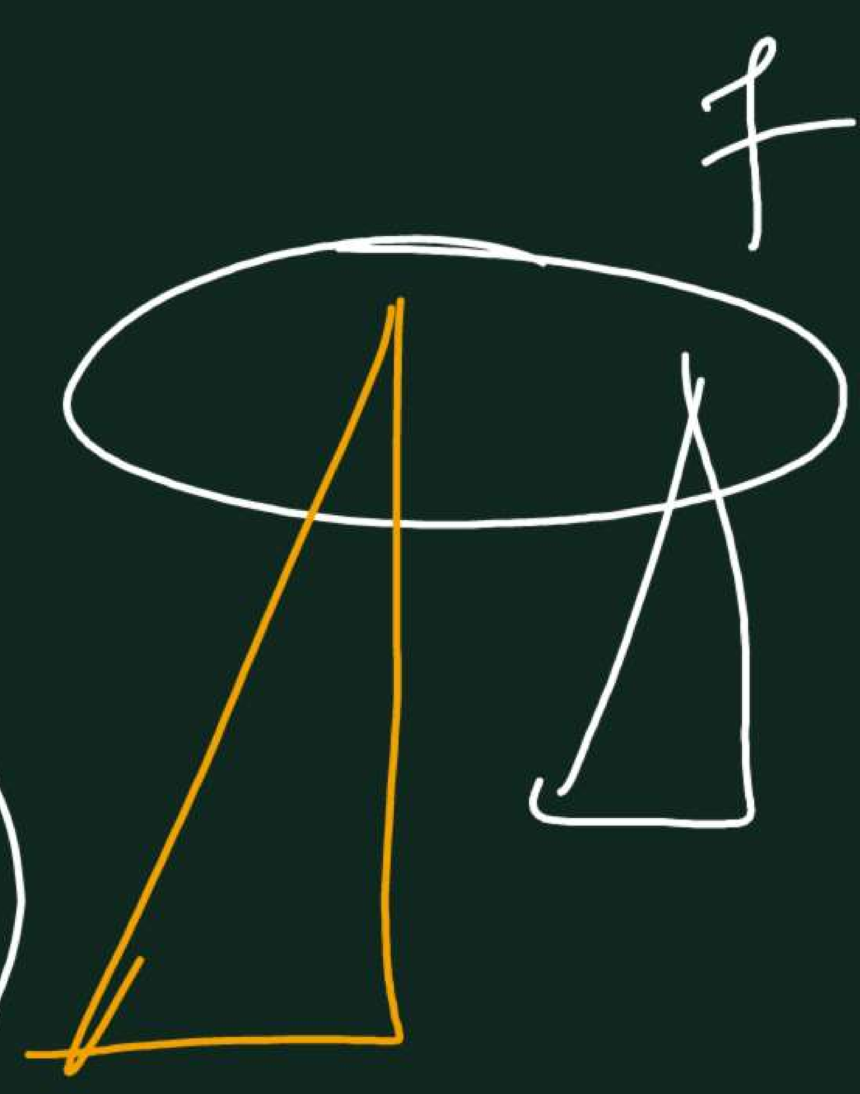
A_6 $\text{heightF}(\text{addTree}(f, t, n)) \equiv \max(\text{hight}(t), \text{hightF}(f))$

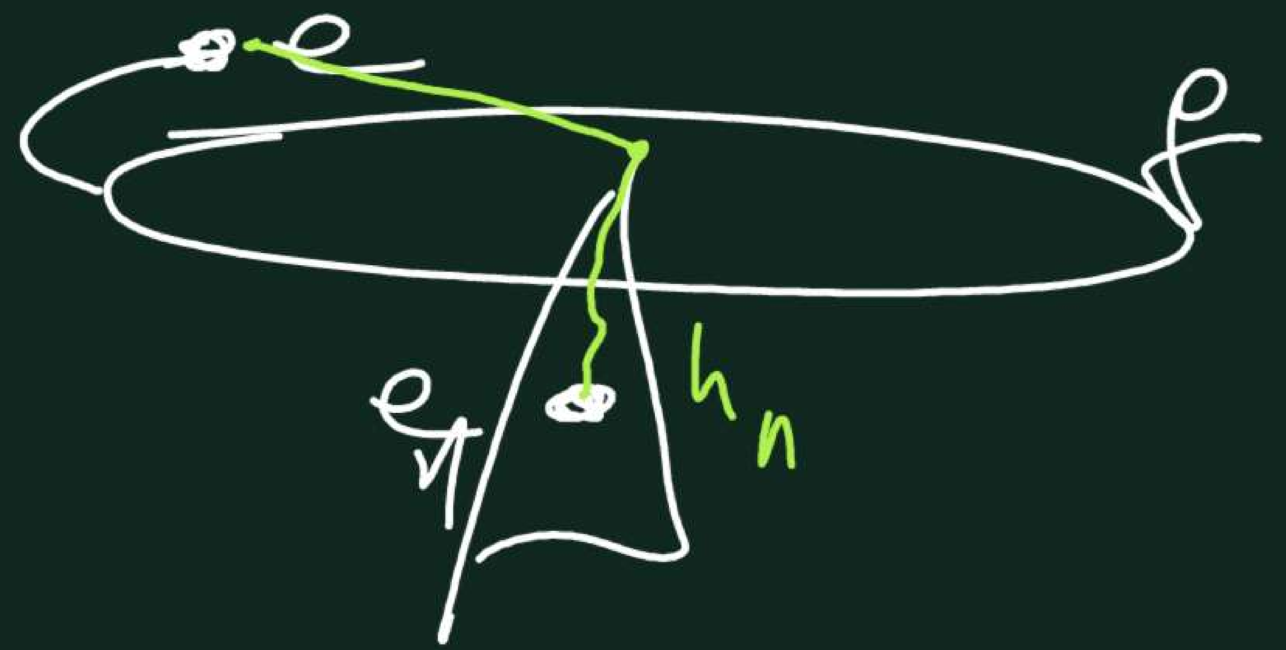
A_7 $PL(\text{new}(e, f)) \equiv PLF(f) + \text{sizeF}(f)$

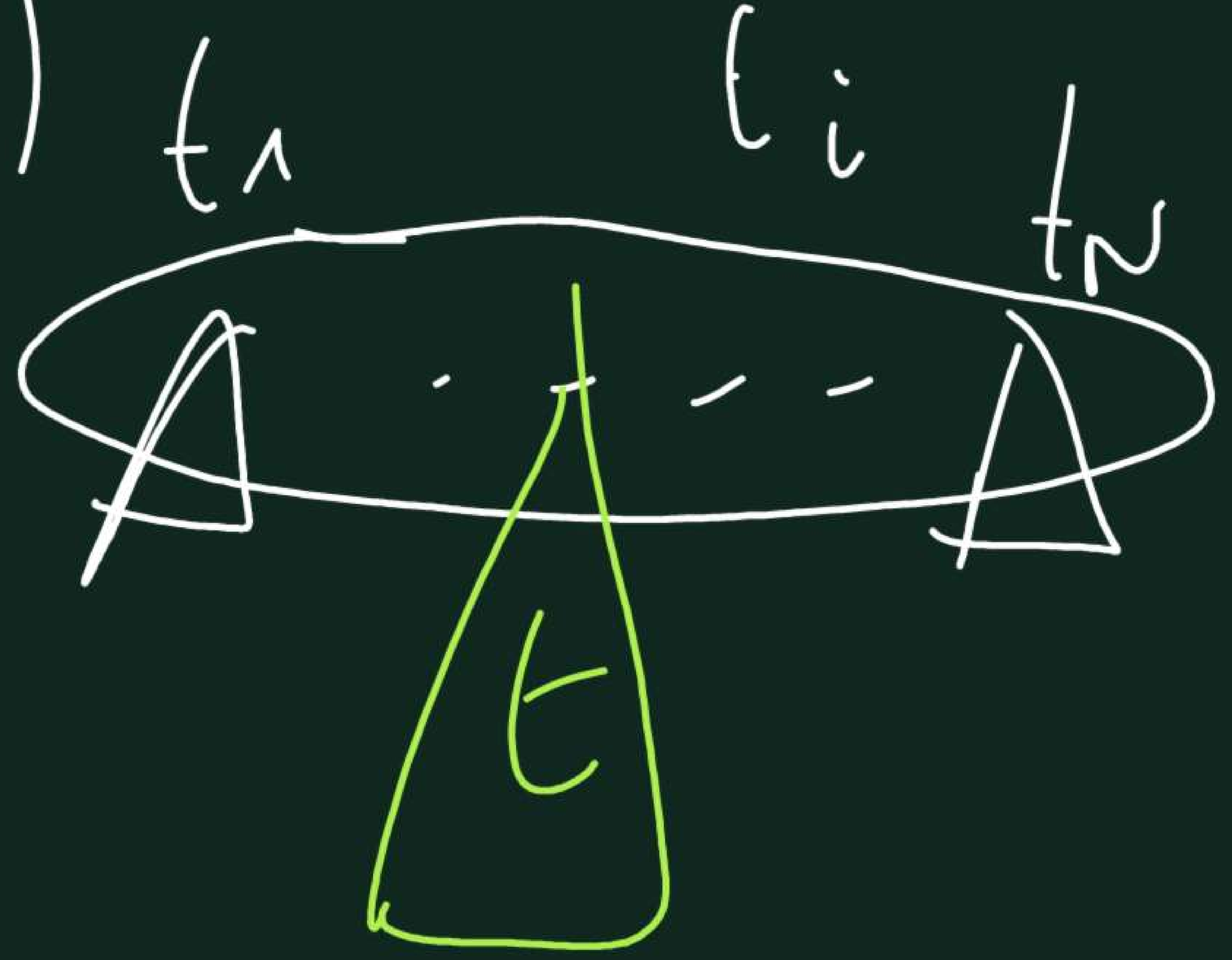
A_8 $PLF(\text{newf}) \equiv 0$

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









Binary trees

def: a BT is

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

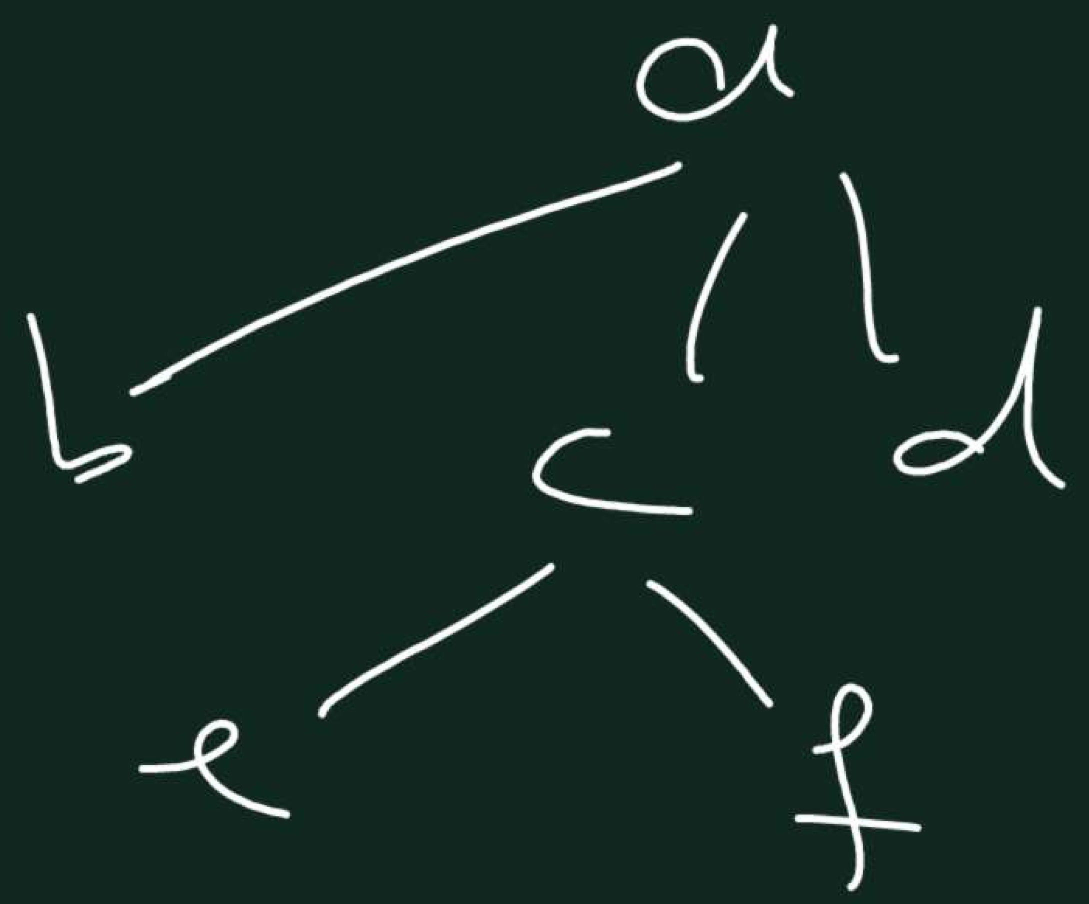
Prop any general tree can be transformed reversibly into a BT without loss of infor/perf

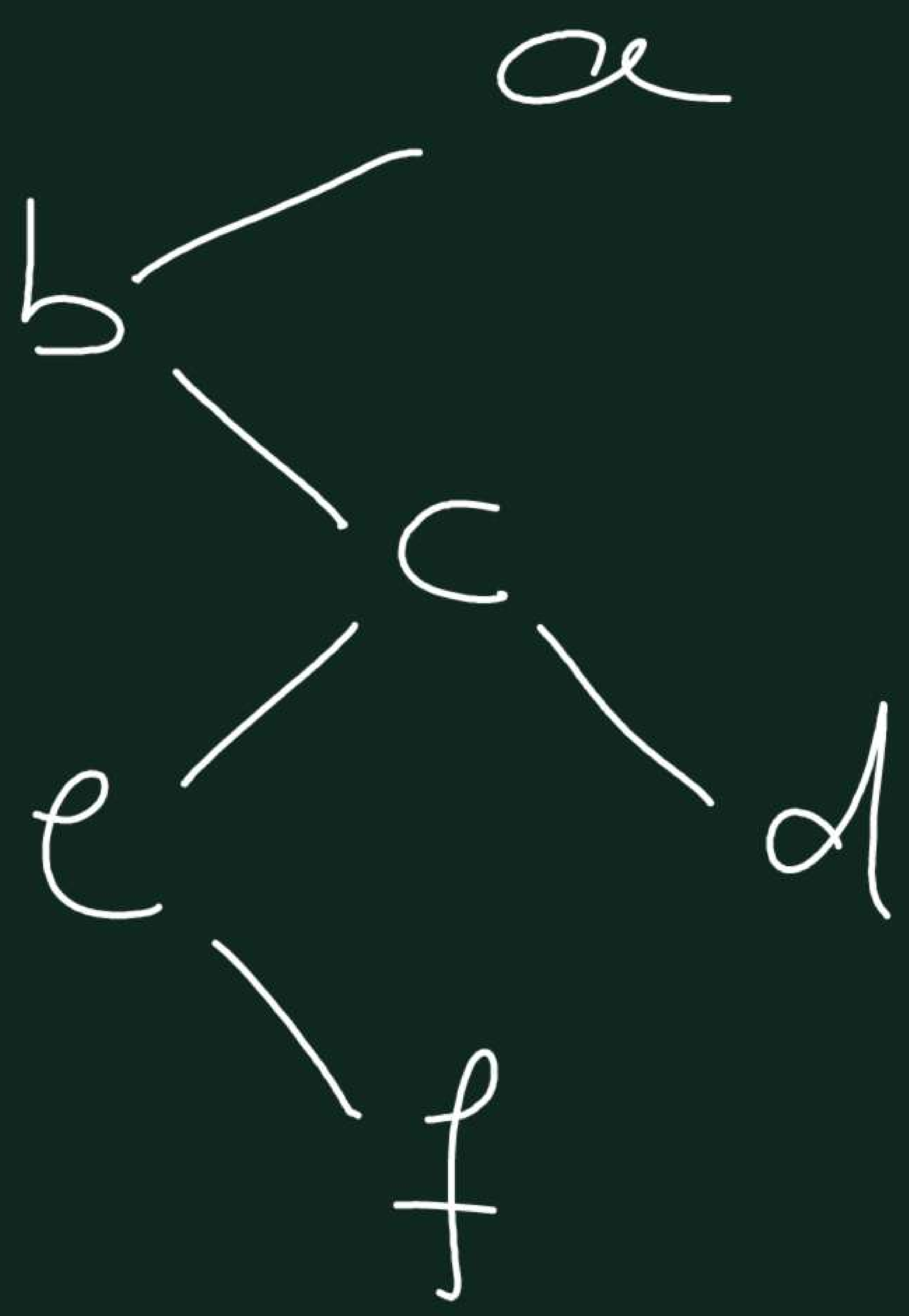
process:

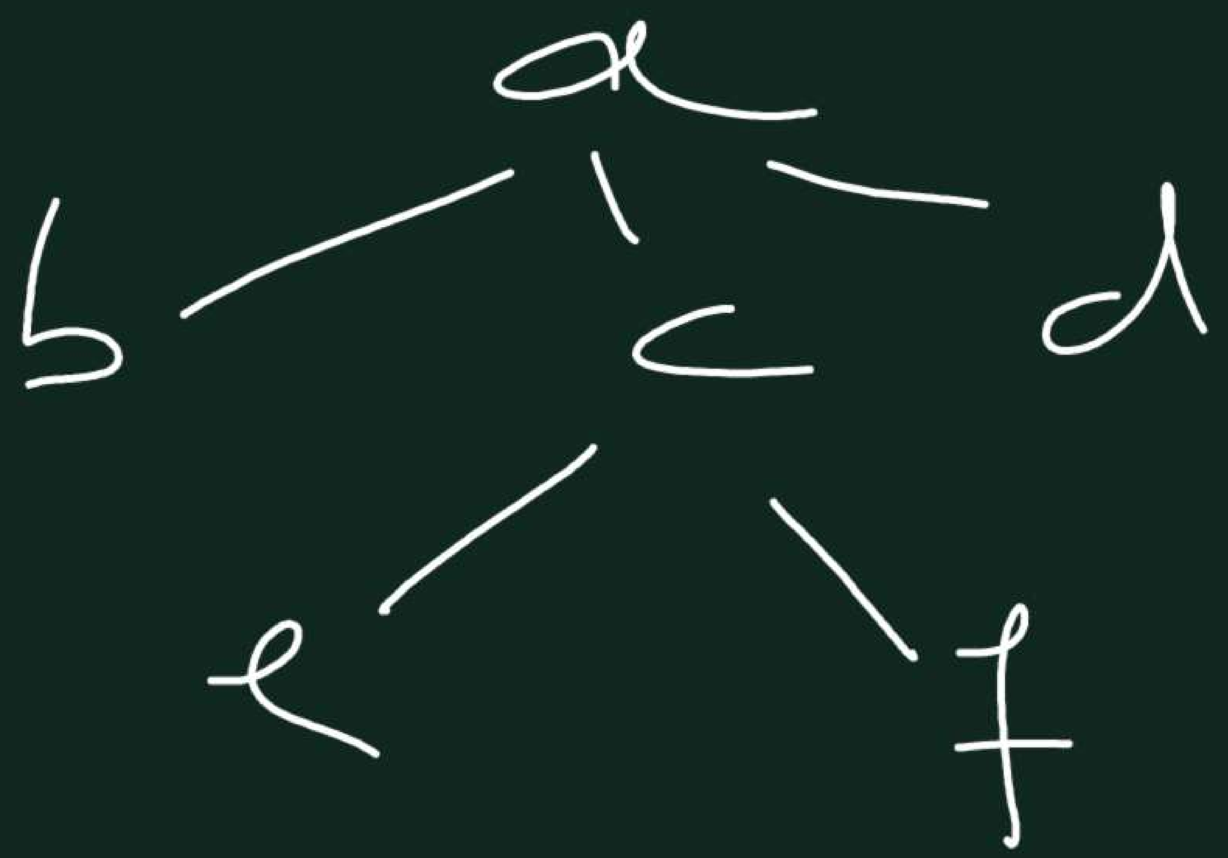
root $\longrightarrow$ root

1st child $\longrightarrow$ left child

right sibling $\longrightarrow$ right child


a
b
c
d
e
f


a
b
c
e
d
f


a
b
c
d
e
f

ADT BT

Use Element, Boolean

Operations

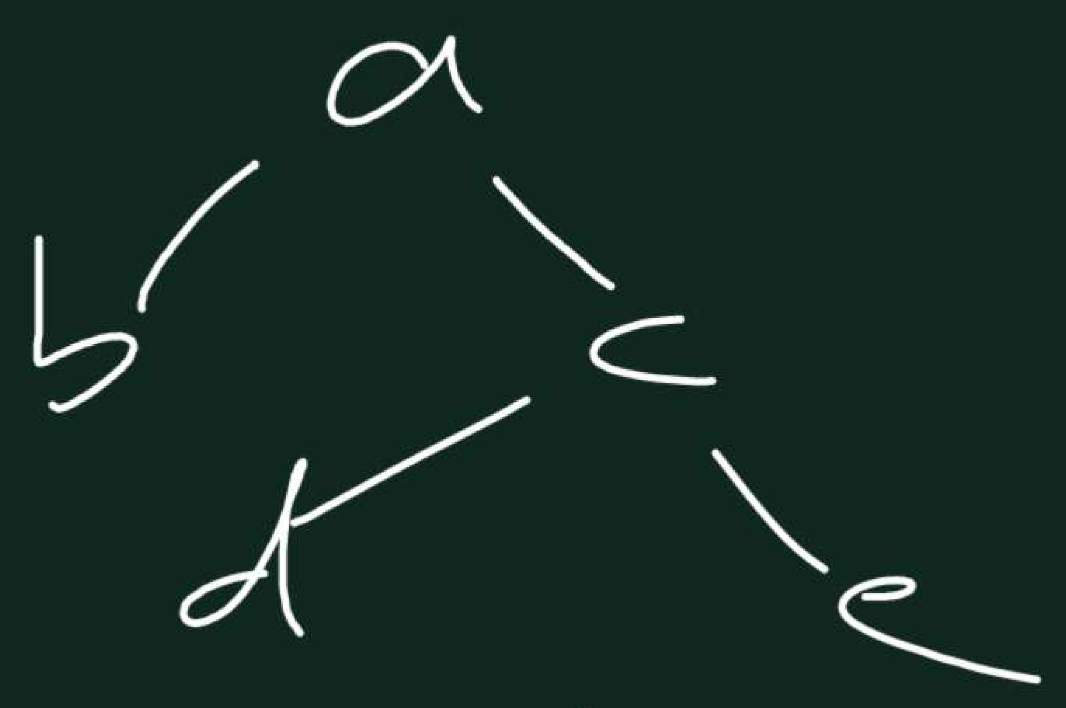
* new : $\longrightarrow$ BT
* $\langle \_, \_, \_ \rangle$: Element $\times$ BT $\times$ BT $\longrightarrow$ BT

root : BT $\longrightarrow$ Element

left : BT $\longrightarrow$ BT

right : BT $\longrightarrow$ BT

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



$\langle a, \langle b, new, new\rangle, \langle c, \langle d, new, new\rangle, \langle e, new, new\rangle\rangle\rangle$

preconditions

root(t) iff t != new

left

right

Axioms

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

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

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

A_4 $\mathrm{contains}(\mathrm{new}, e) \equiv F$

A_5 $\mathrm{contains}(\langle e, L, R\rangle, e) \equiv T$

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

ADT extension BT

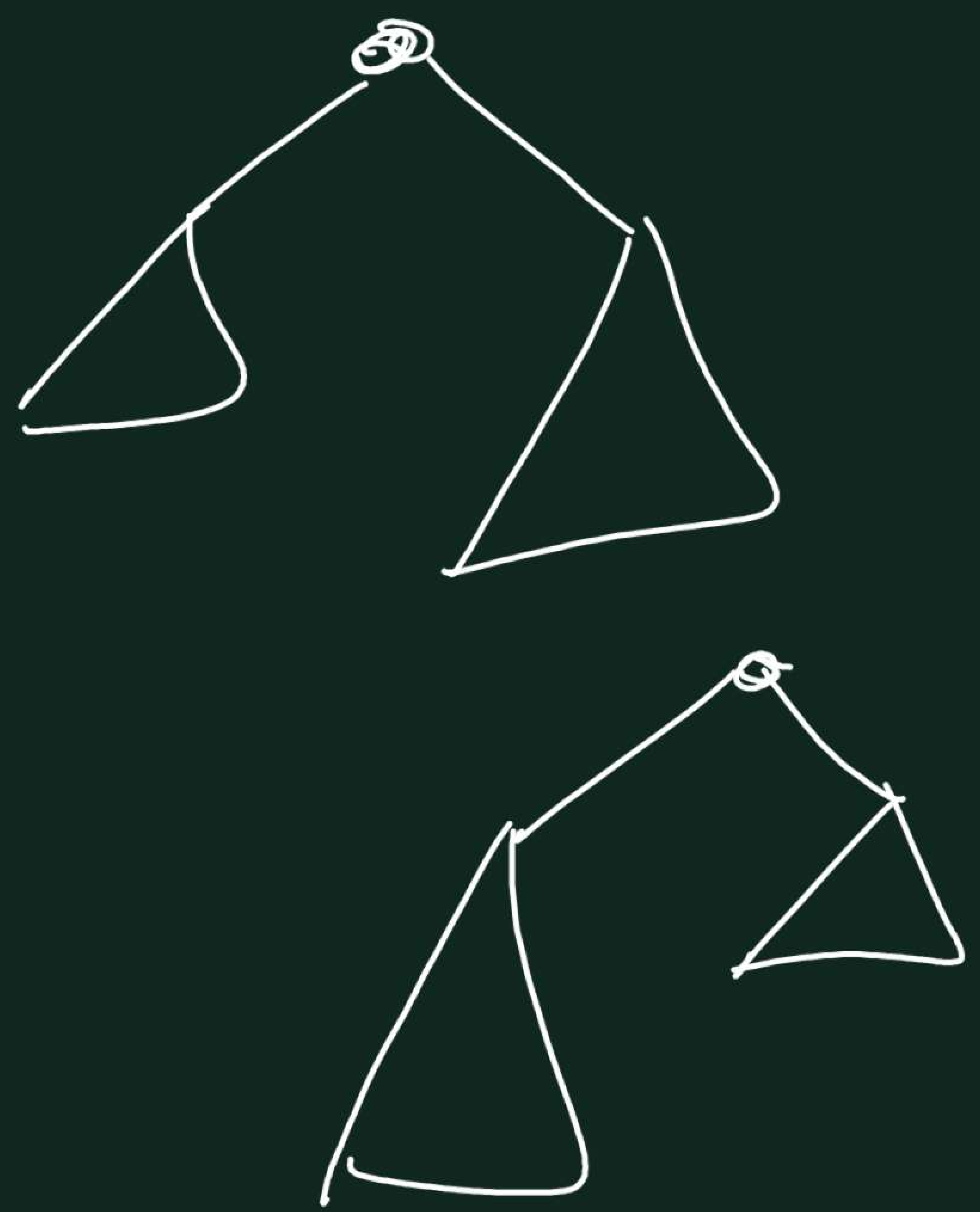
Use Integer

Operations

size : BT $\longrightarrow$ Integer

height : BT $\longrightarrow$ Integer

PL : BT $\longrightarrow$ Integer



Axioms

A_1 $\text{size}(\text{new}) \equiv 0$

A_2 $\text{size}(\langle r, L, R \rangle) \equiv 1 + \text{size}(L) + \text{size}(R)$

A_3 $\text{height}(\text{new}) \equiv -1$

A_4 $\text{height}(\langle r, L, R \rangle) \equiv 1 + \max(\text{height}(L), \text{height}(R))$

$A_5 \quad PL(new) \equiv 0$

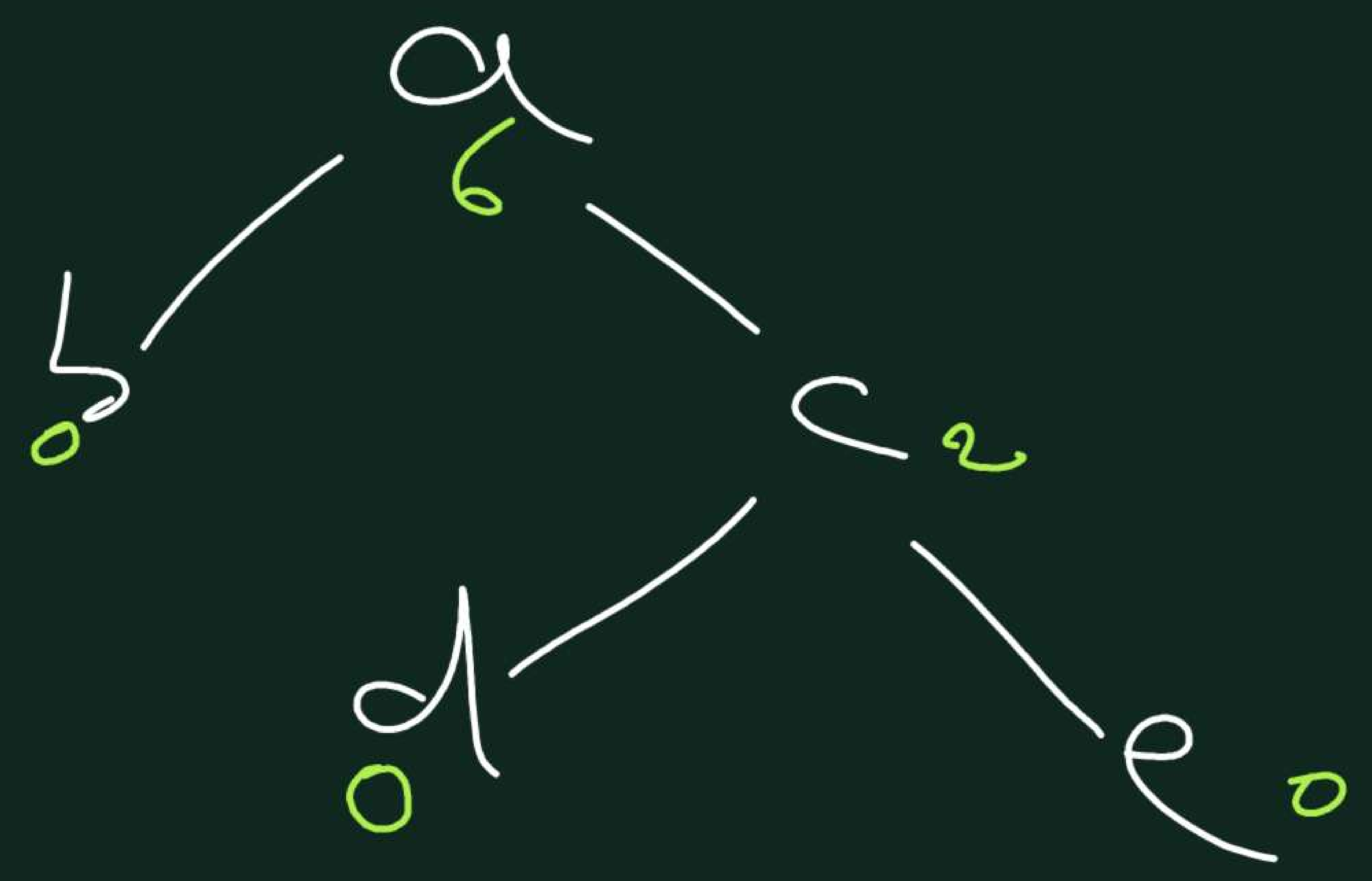
$A_6 \quad PL(\langle r, L, R\rangle) \equiv PL(L) + PL(R) + \begin{cases} size(L) + size(R \\ size(\langle r, L, R\rangle) - 1 \end{cases}$

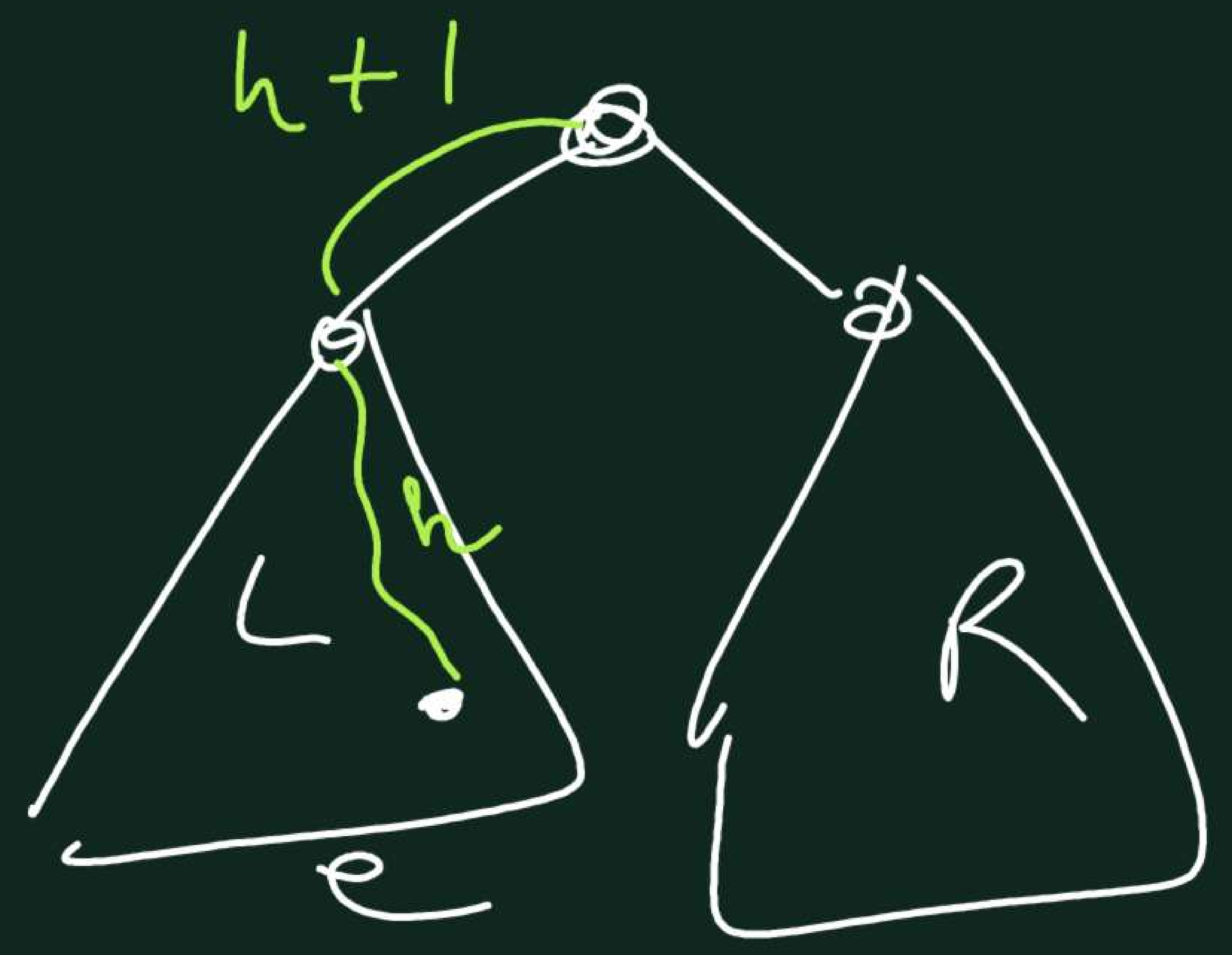
1×0

2×1

2×2

6





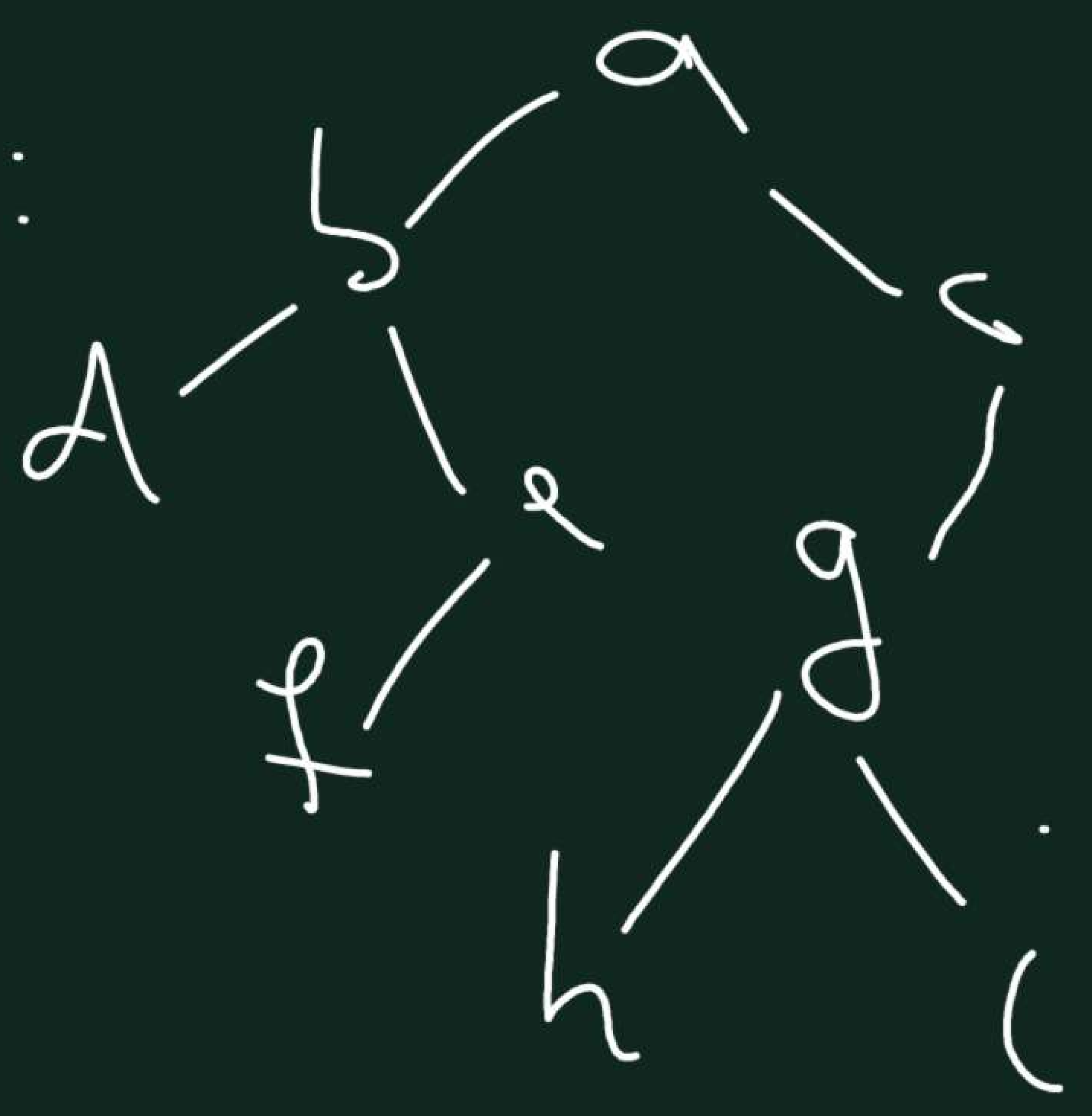
Listing of the elements of a BT (tree traversal)

2 main categories:

1) breadth-first: list all elements at depth n before listing elements at depth $n+1$

2) depth-first: list all elements of one child before listing any element of the other

ex:

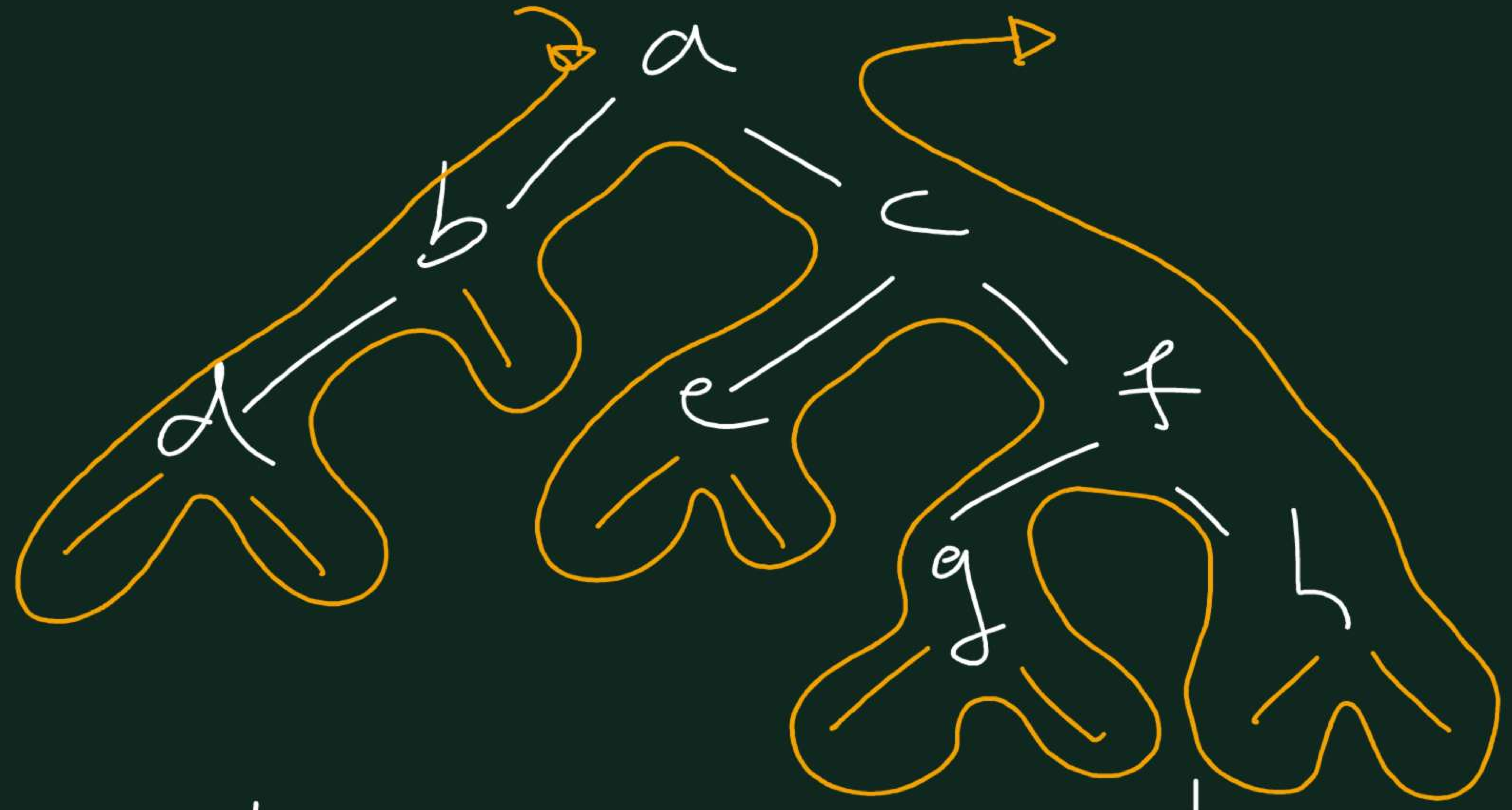


BF, $1 \to n$: a b c d e g f h i

DF, $l \to r$: a b d e f c g h i

We visit each node up to 3 times
→ visit1, visit2, visit3

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



pre: a b d c e f g h
in: d b a e c g f h
post: d b e g h f c a

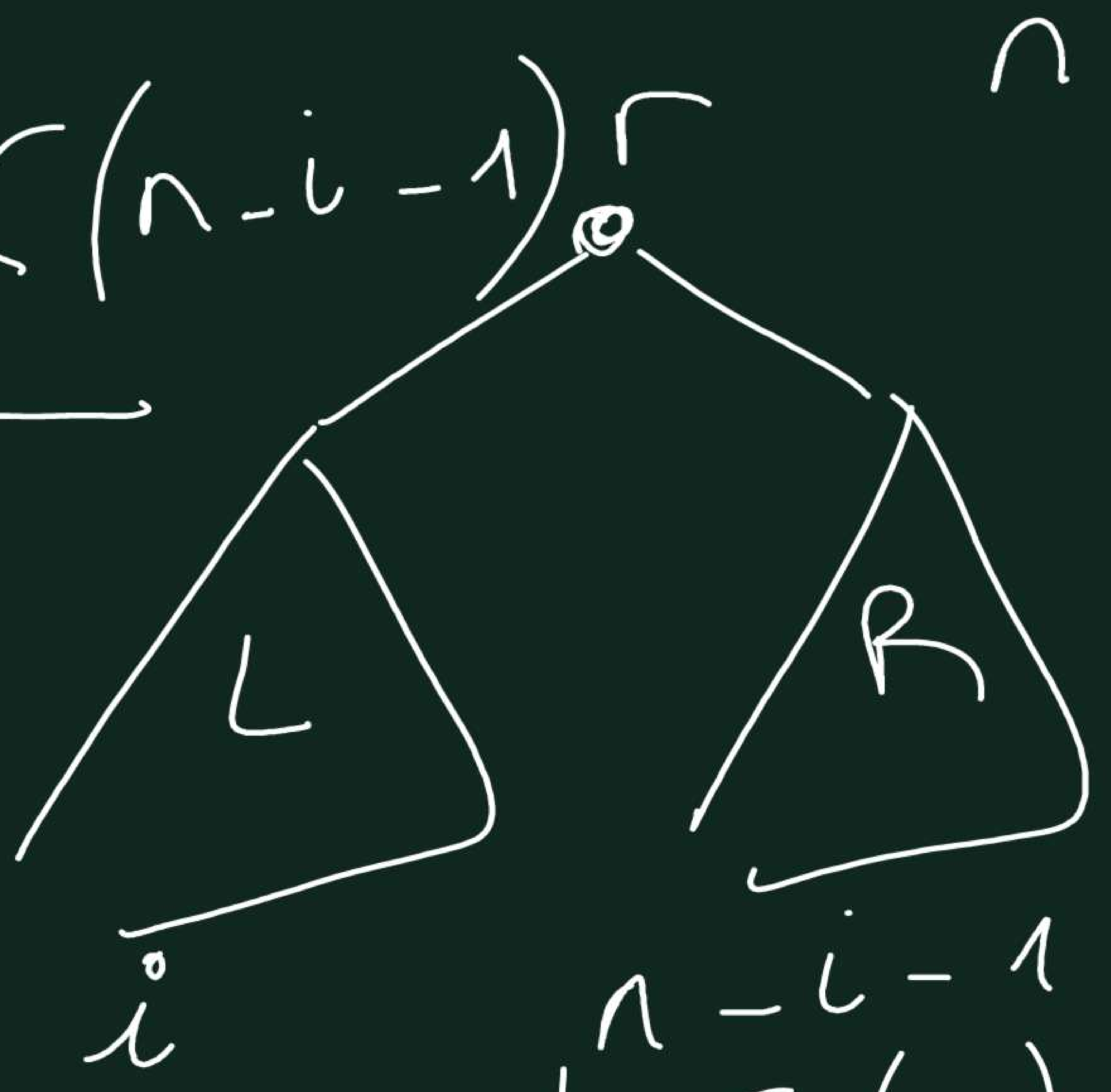
Preorder: visit2 = visit3 = $\emptyset$
inorder: visit1 = visit3 = $\emptyset$
postorder: visit1 = visit2 = $\emptyset$

```
preorder(t) {
    if (t != null) {
        print(root(t))
        preorder(left(t))
        postorder(right(t))
    }
}
```

Complexity?

param: n, size of t
unit op: print

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



guess: $C(n) = n$?

prop: $C(n) = n$

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

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

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

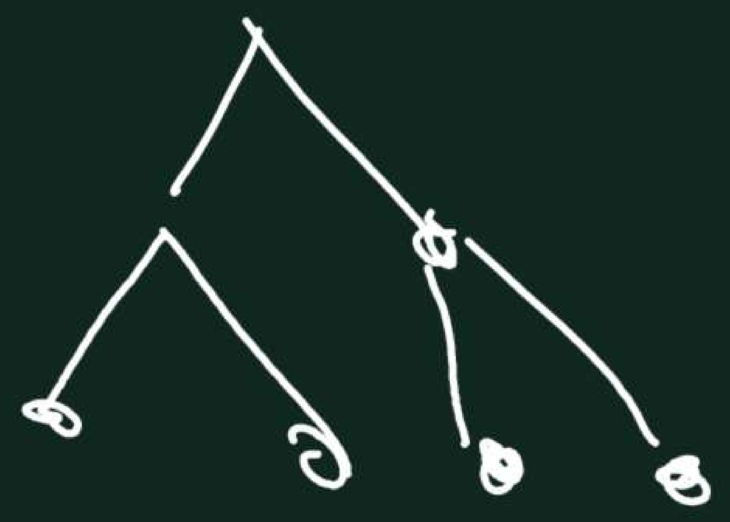
n	$C(n)$
1	1
2	2
3	3
⋮	
n	n

examples/exercises on induction:

What is the maximum number of leaves for a BT of size N?

What is the minimum size of a BT having N leaves?

N_L	N
1	1
2	3
3	5
4	7

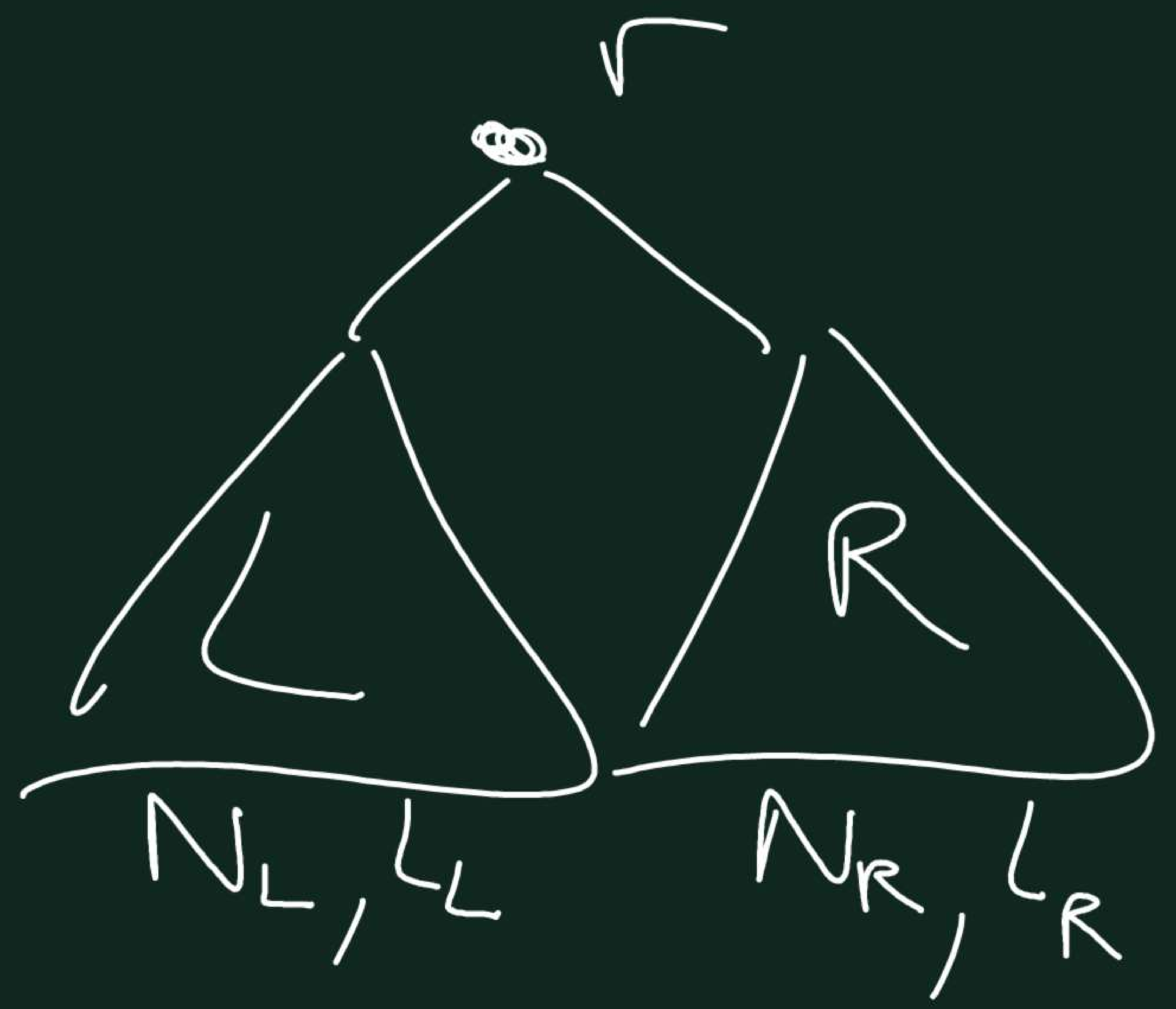


$N = 2N_L - 1$?

$p(1) = p(2) = p(3) = T$

$N = 2L - 1$?

N = size of t
L = number of leaves in t



$$\begin{cases} N = N_L + N_R + 1 \\ L = L_L + L_R \\ N_L = 2L_L - 1 \\ N_R = 2L_R - 1 \end{cases}$$

$$N = 2L_L \cancel{-1} + 2L_R - 1 \cancel{+1}$$
$$= 2L - 1 \quad \text{qed}$$

t a BT of size N. Number of links in t?

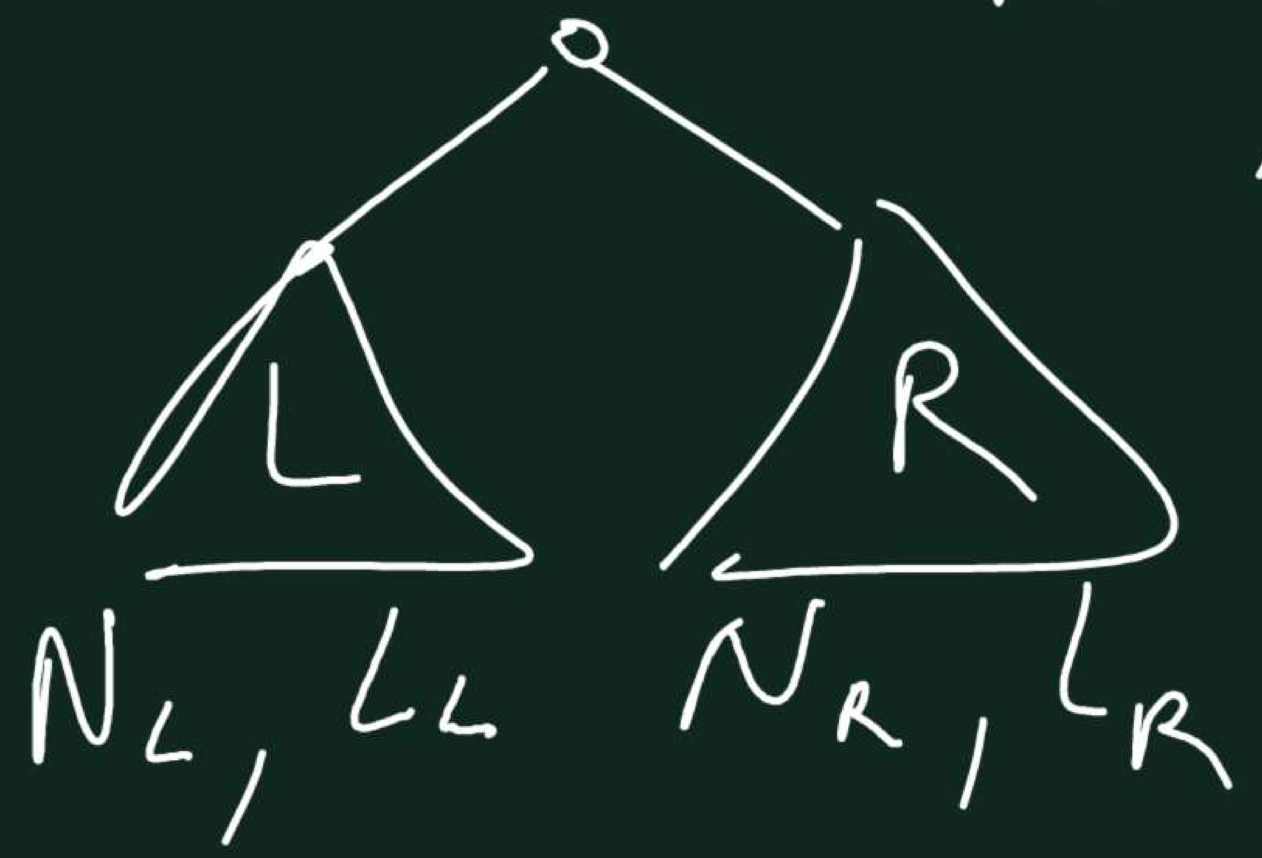
guess: $L = N - 1$

$P(1) = P(2) = P(3) = T$

$N \quad , \quad L$

hyp: $\forall i < n, \; L_i = i - 1$

$$\begin{cases} N = N_L + N_R + 1 \\ L = L_L + L_R + 2 \\ L_L = N_L - 1 \\ L_R = N_R - 1 \end{cases}$$



$$\begin{aligned} L &= N_L \cancel{- 1} + N_R \cancel{- 1 + 2} \\ &= N_L + N_R \\ &= N - 1 \end{aligned}$$

qed

Binary search trees (BST)

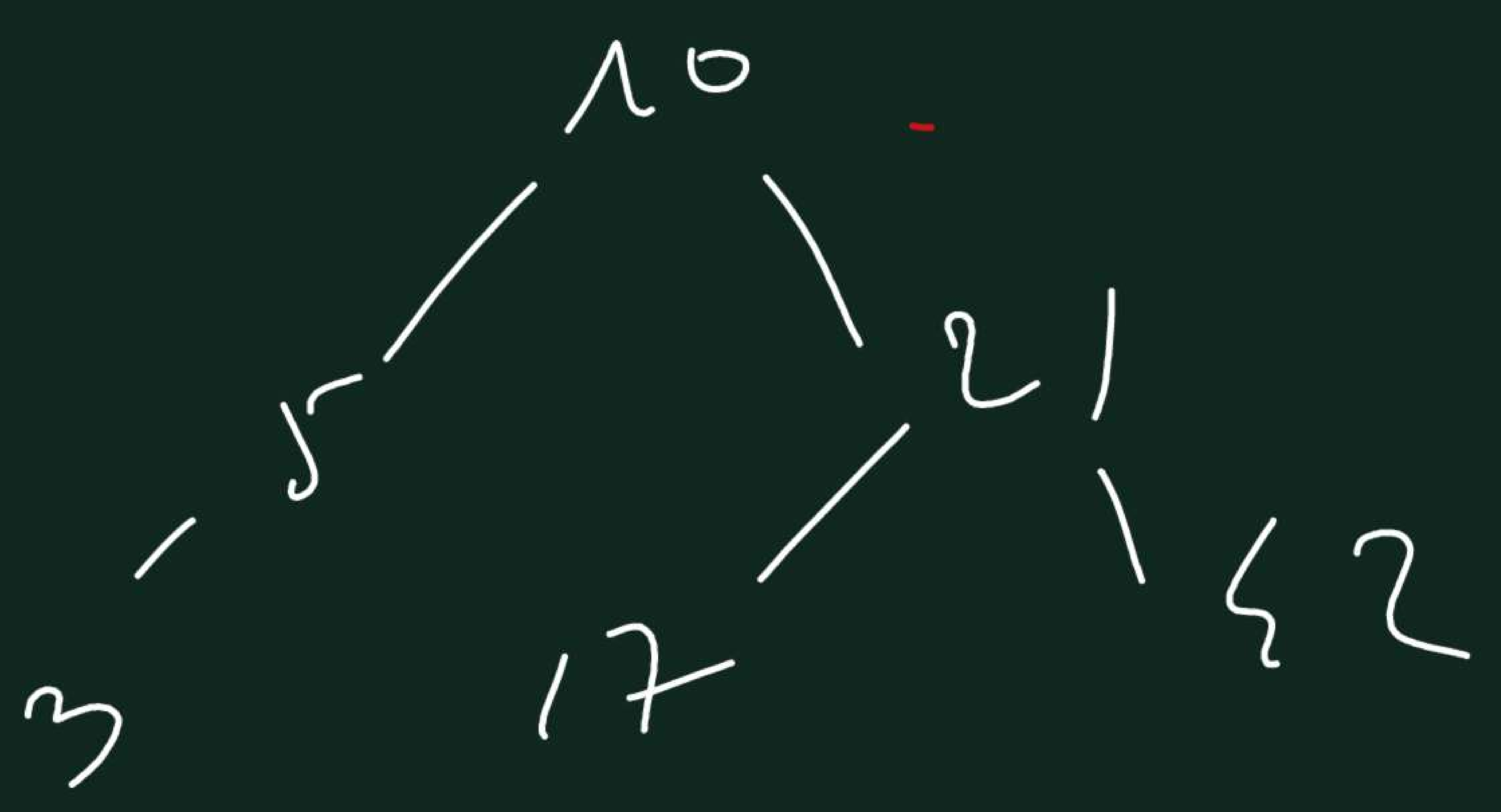
(needs a "$\leq$" operation on elements)

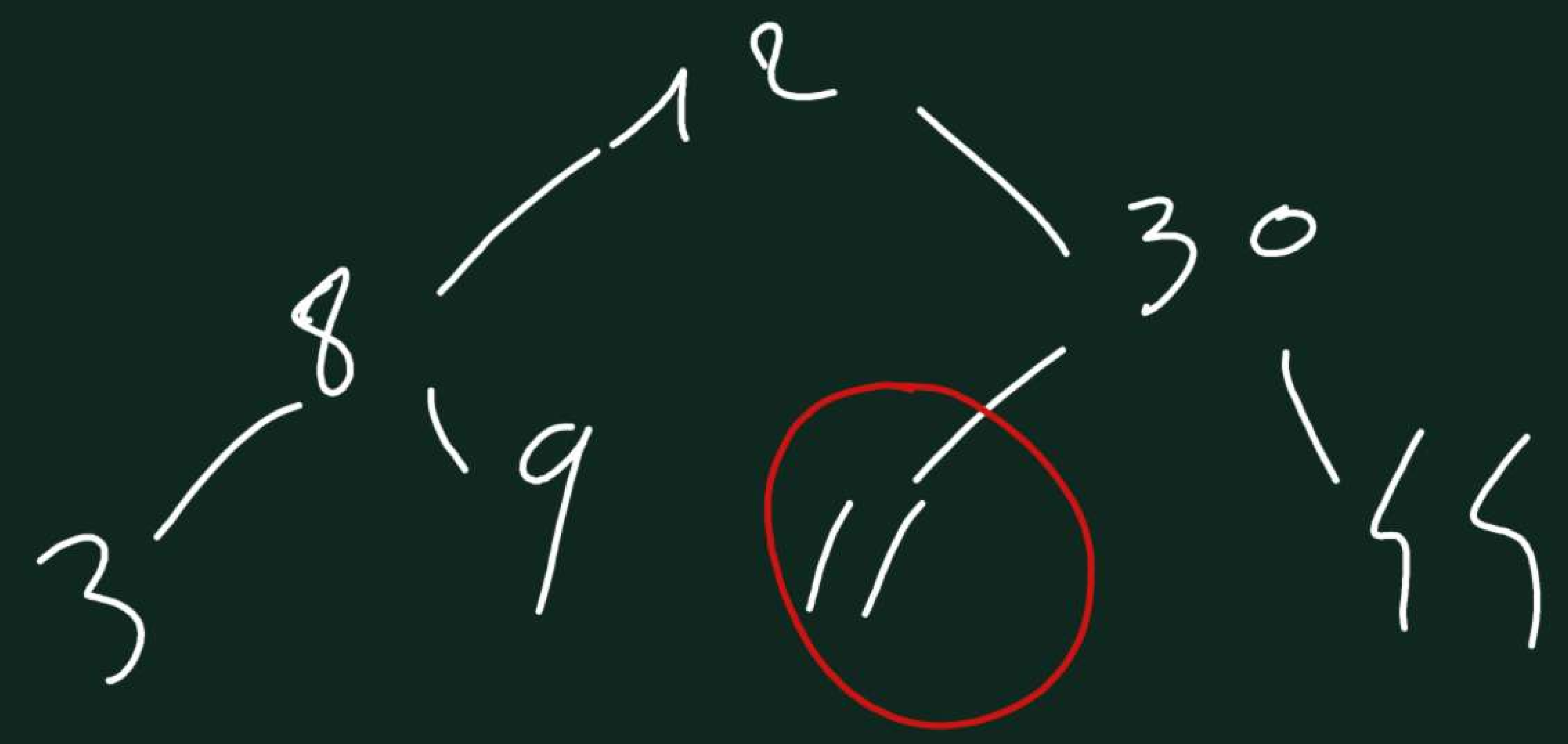
<u>def</u>: a BST is a BT /

$\text{root} \leq e$ if $e \in \text{right}(t)$

$e \leq \text{root}$ if $e \in \text{left}(t)$

$\text{left}(t)$ and $\text{right}(t)$ are BSTs





ADT BST extends BT

Use Element ($\leq$)

Operations

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

add : BST $\times$ Element $\longrightarrow$ BST

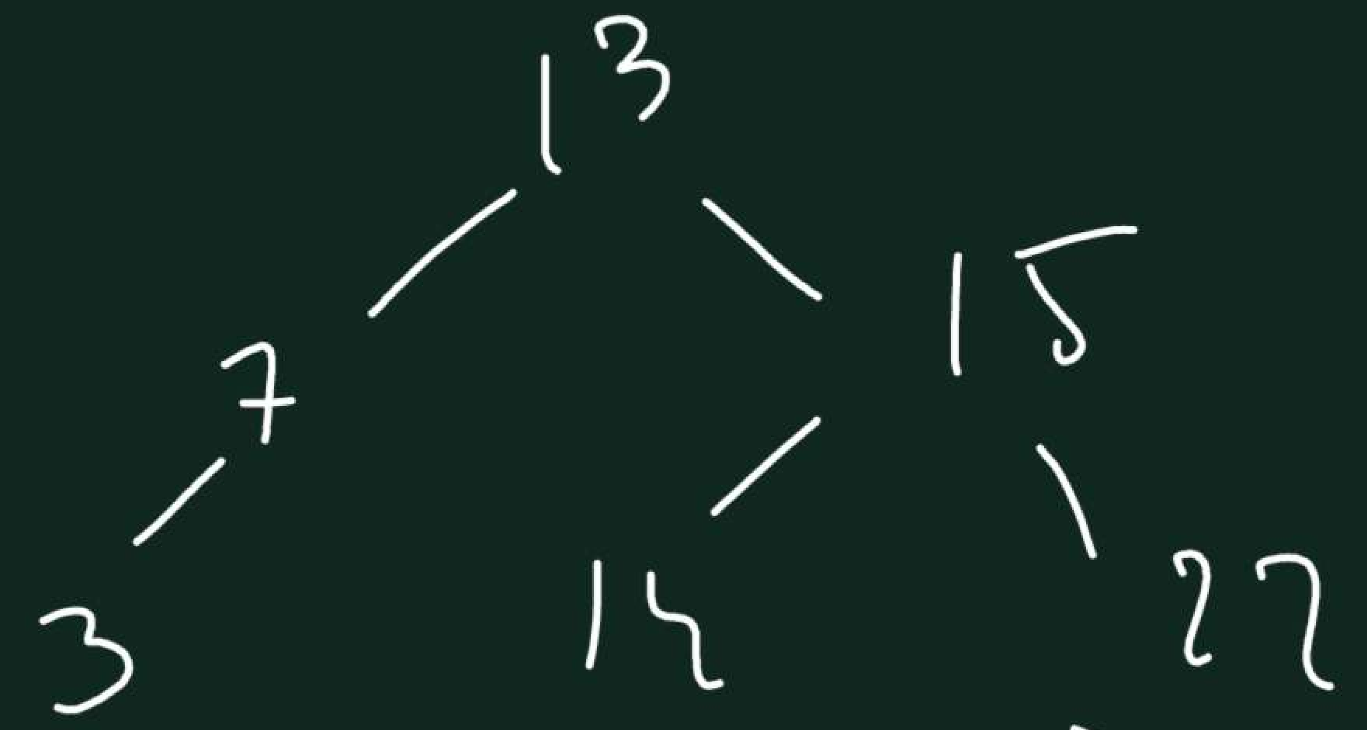
Axioms

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

A_2 contains(<r, L, R>, r) $\equiv$ T

A_3 $e \leq r \Rightarrow$ contains(<r, L, R>, e) $\equiv$ contains(L, e)

A_4 contains(<r, L, R>, e) $\equiv$ contains(R, e)



contains(13, 15)

A_4 contains(15, 15)

A_2 T

contains(13, 8)

A_3 contains(7, 8)

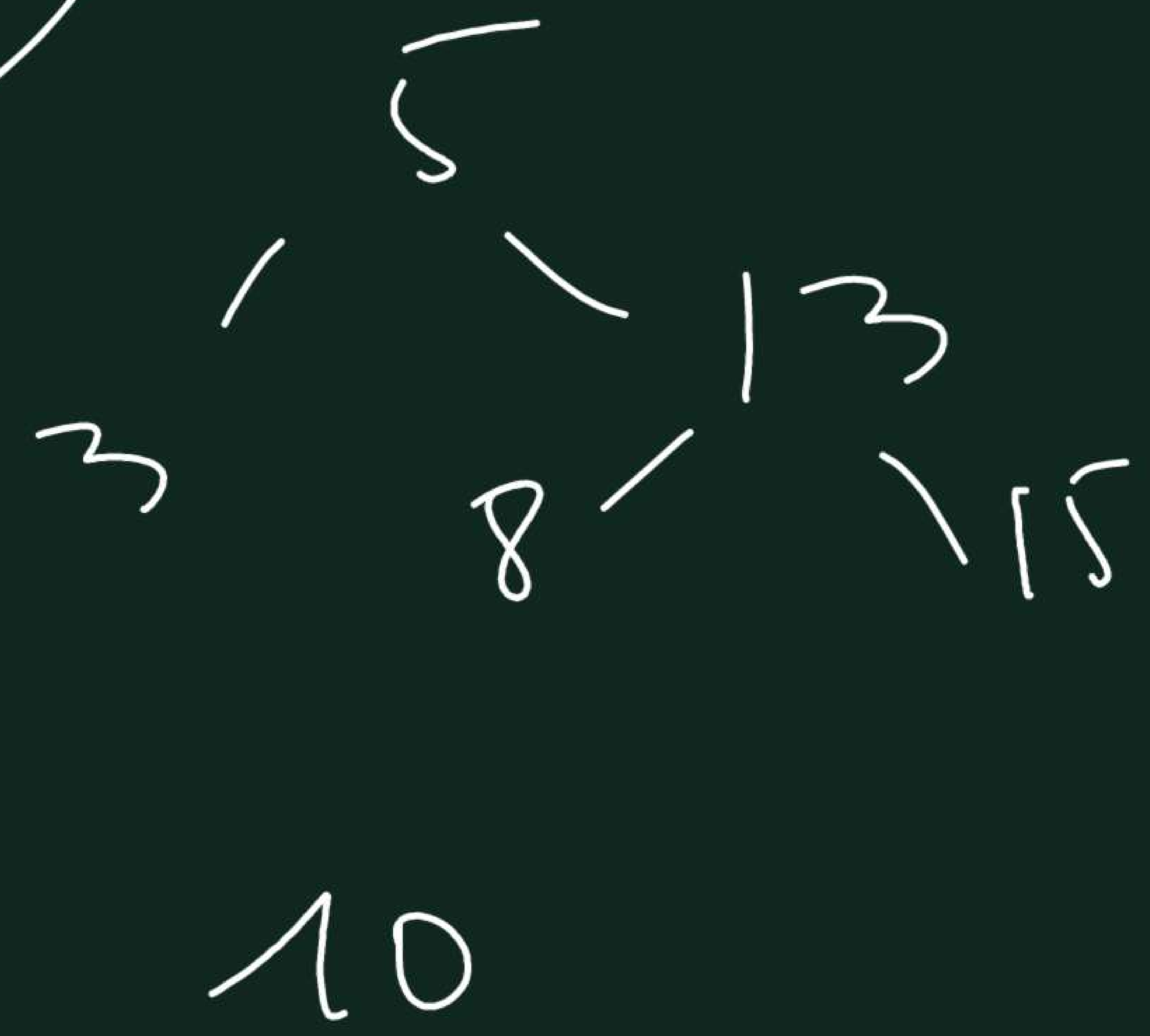
~~A_4~~ contains(new, 8)

A_1 F

$A_5 \quad add(new, e) \equiv \langle e, new, new \rangle$

$A_6 \quad e \leq r \implies add(\langle r, L, R \rangle, e) \equiv add(L, e)$

$A_7 \quad add(\langle r, L, R \rangle, e) \equiv add(R, e)$



$add(5, 10)$

$A_7 \quad add(13, 10)$

$A_6 \quad add(8, 10)$

$\cancel{A_7} \quad add(new, 10)$

$A_5 \quad \langle 10, new, new \rangle$

WRONG!

$A_5 \quad add(new, e) \equiv \langle e, new, new \rangle$

$A_6 \quad e \leq r \Rightarrow (add\langle r, L, R \rangle, e) \equiv \langle r, add(L, e), R \rangle$

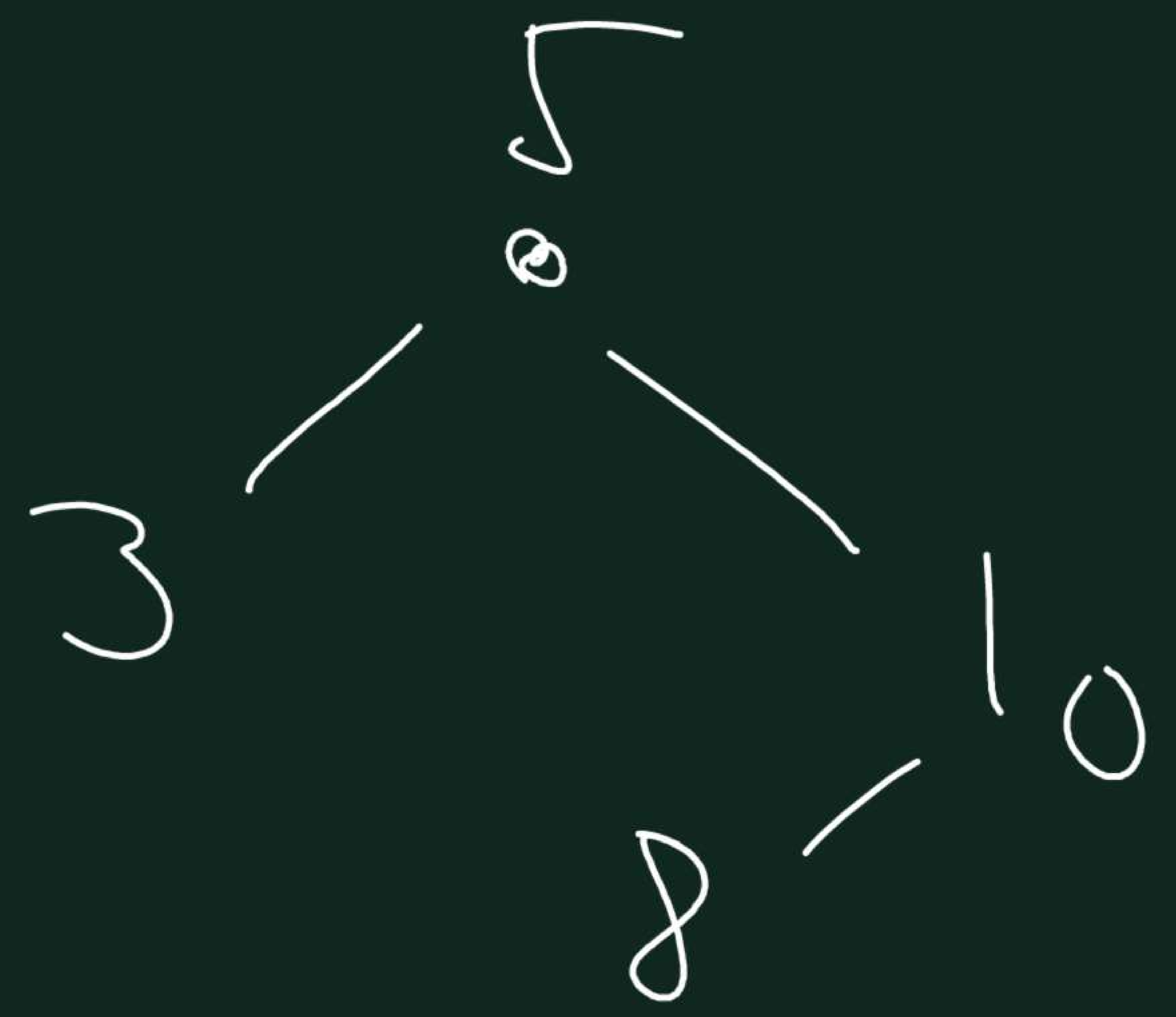
$A_7 \quad add(\langle r, L, R \rangle, e) \equiv \langle r, L, add(R, e) \rangle$

$add(5, 8)$

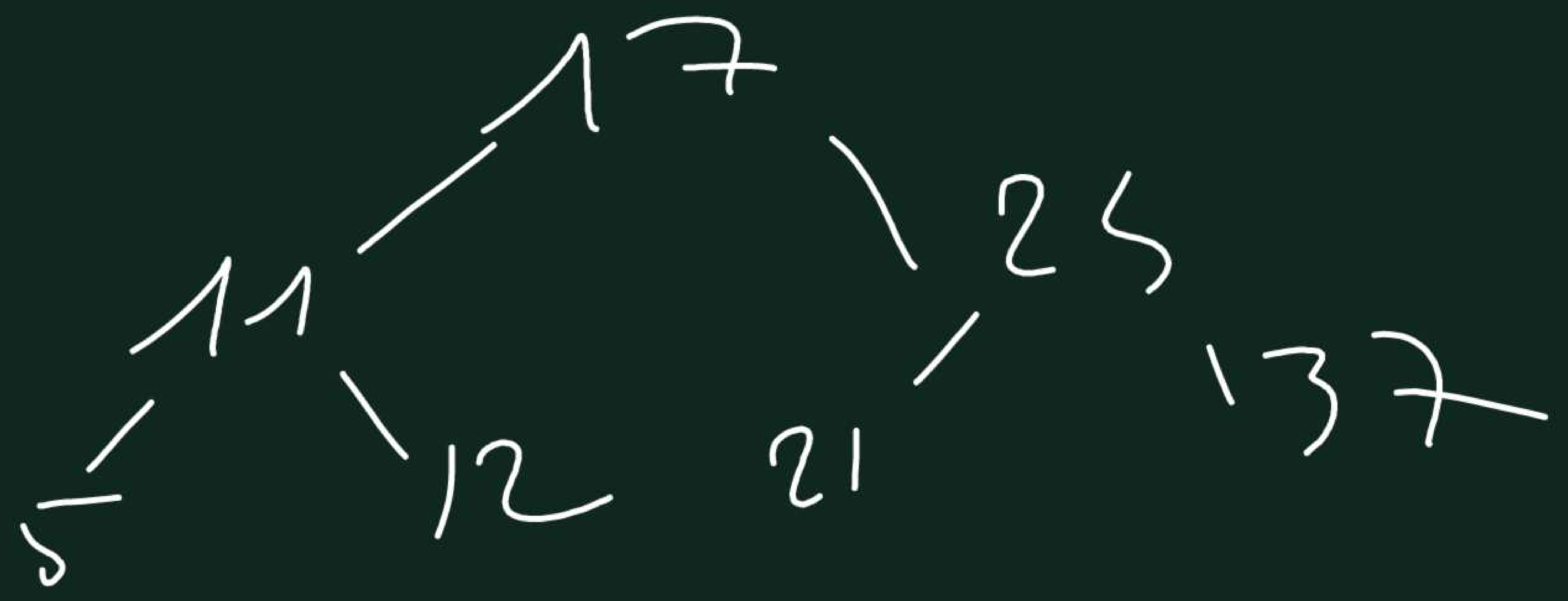
$A_7 \quad \langle 5, \langle 3 \rangle, add(10, 8) \rangle$

$A_6 \quad \langle 5, \langle 3 \rangle, \langle 10, add(new, 8), new \rangle \rangle$

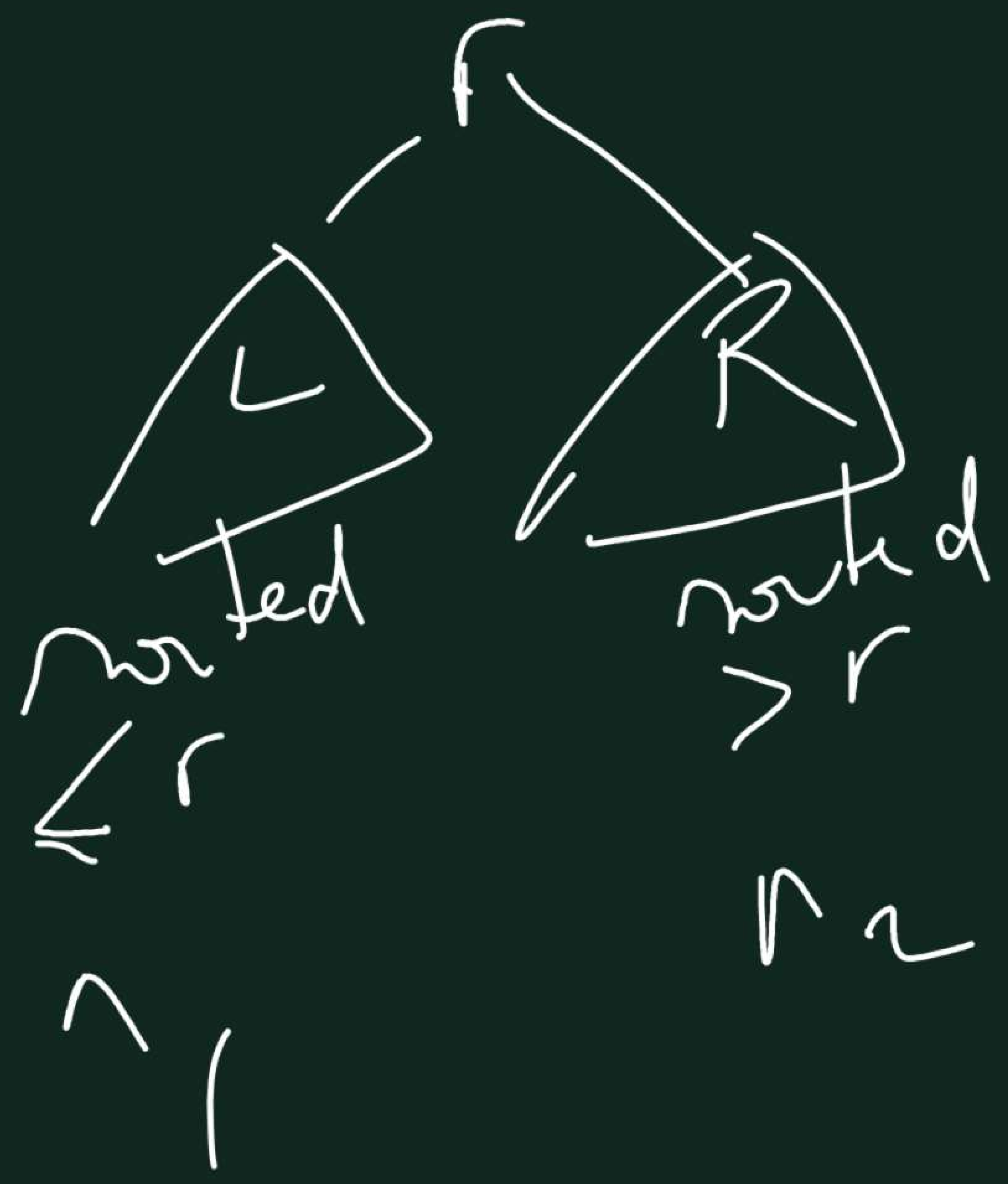
$A_5 \quad \langle 5, \langle 3 \rangle, \langle 10, \langle 8 \rangle, new \rangle \rangle$



Listing elements of a BST



Pre	17	11	5	12	25	21	37
In	5	11	12	17	21	25	37
Post	5	12	11	21	37	25	17



$$\underbrace{e_1 \cdots e_{n_1}}_{\substack{\text{sorted} \\ < r}} \; r \; \underbrace{e_{n_1+1} \cdots e_{n_1+n_2}}_{\substack{\text{sorted} \\ > r}}$$

$O(N \log N)$

→ inorder

N

$$\sum_{i=1}^{N} \log i \le \sum_{i=1}^{N} \log N \le \log N \times N$$