



Binary Search Trees Class-1

Special class

target = 2

→ Binary Search

2	6	10	15	18	20	22
0	1	2	3	4	5	6

2	6	10
0	1	2

2

T.C → $O(\log n)$



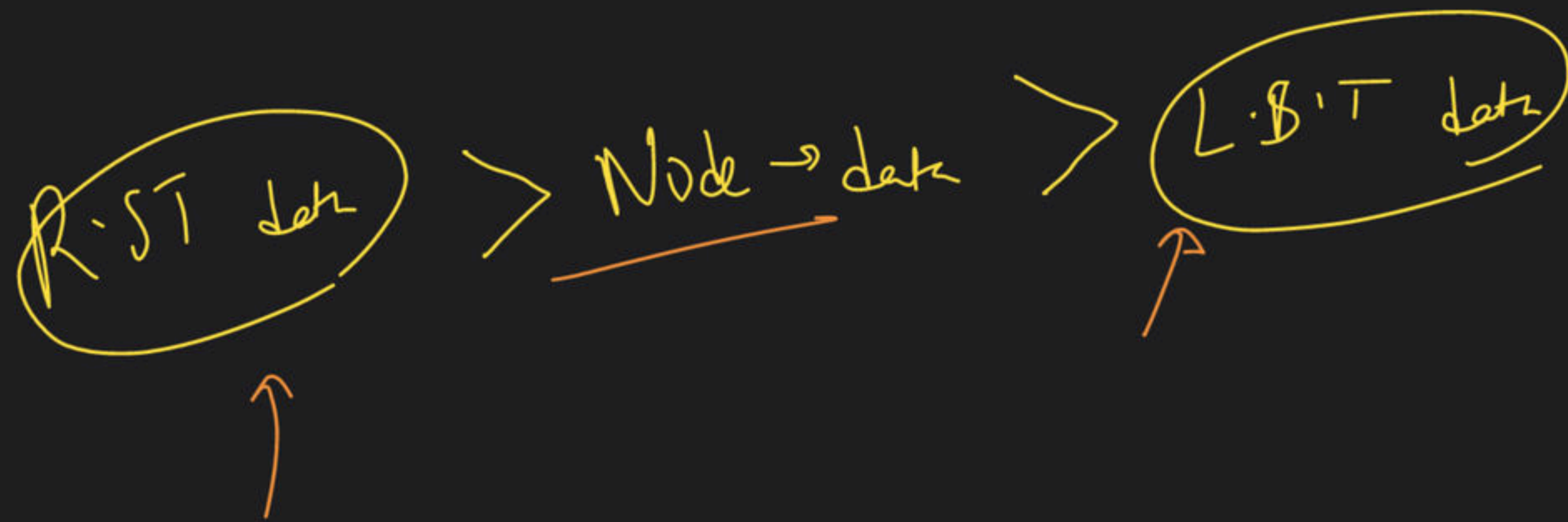
* node \rightarrow property



left subtree

right subtree



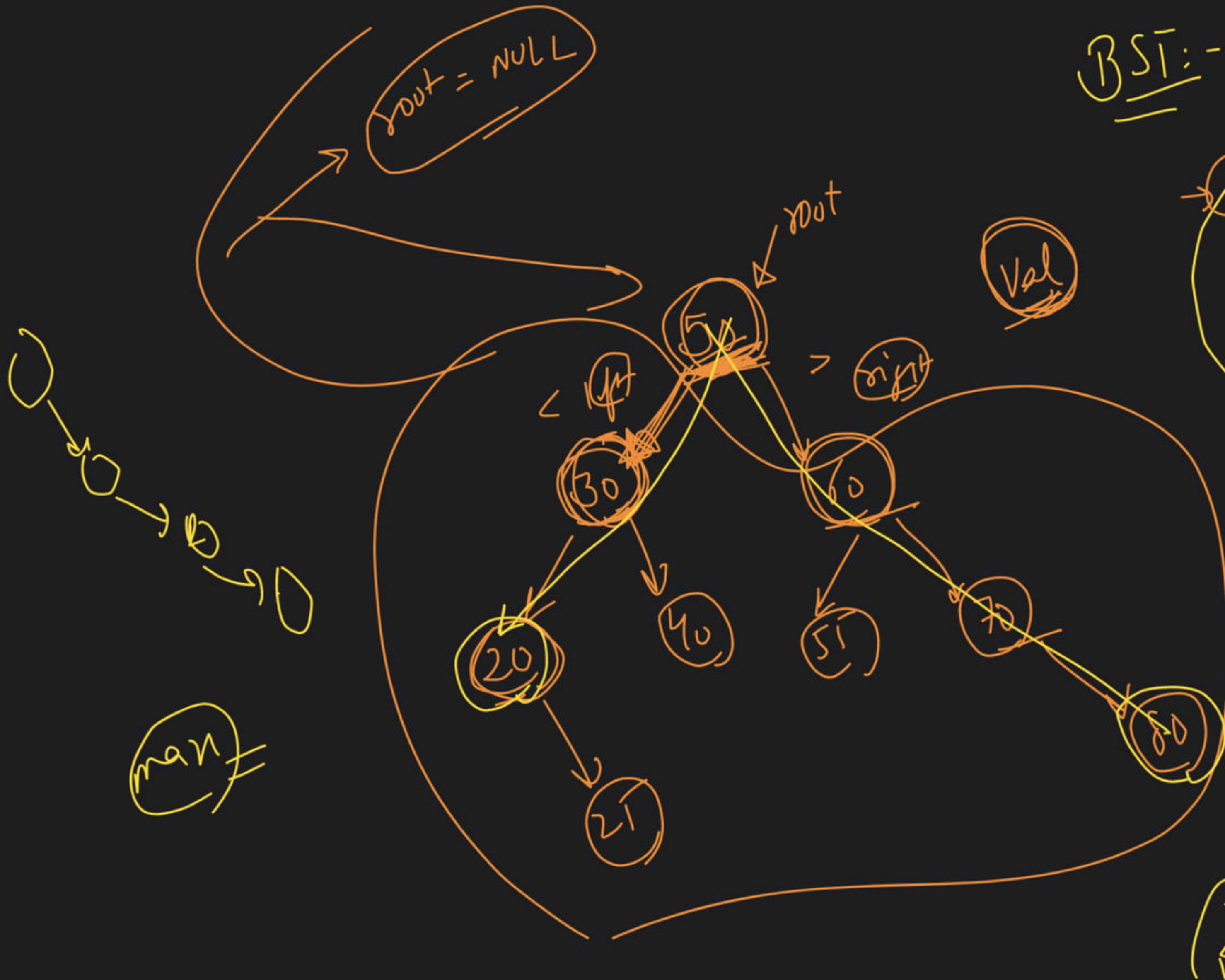


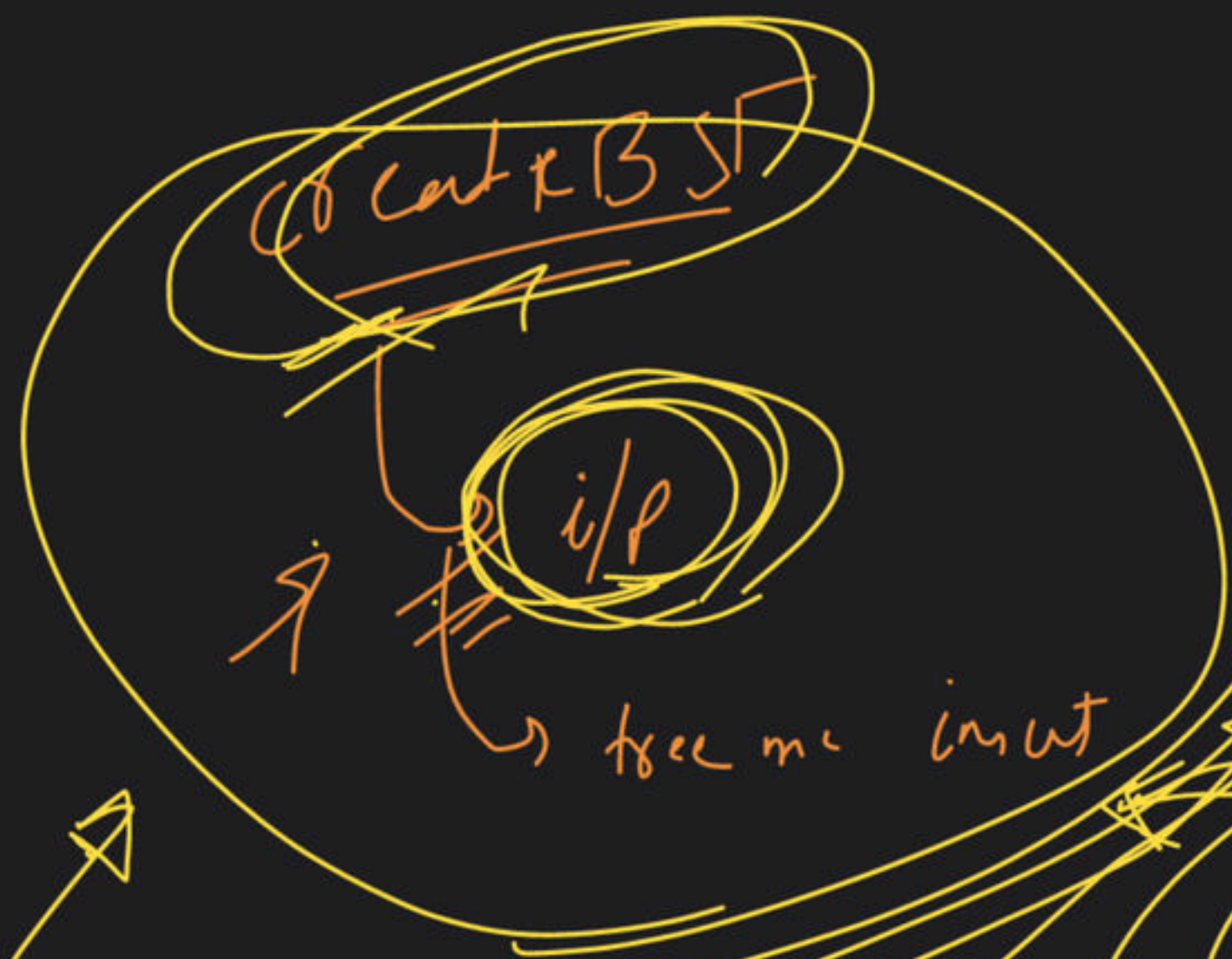
BST:-



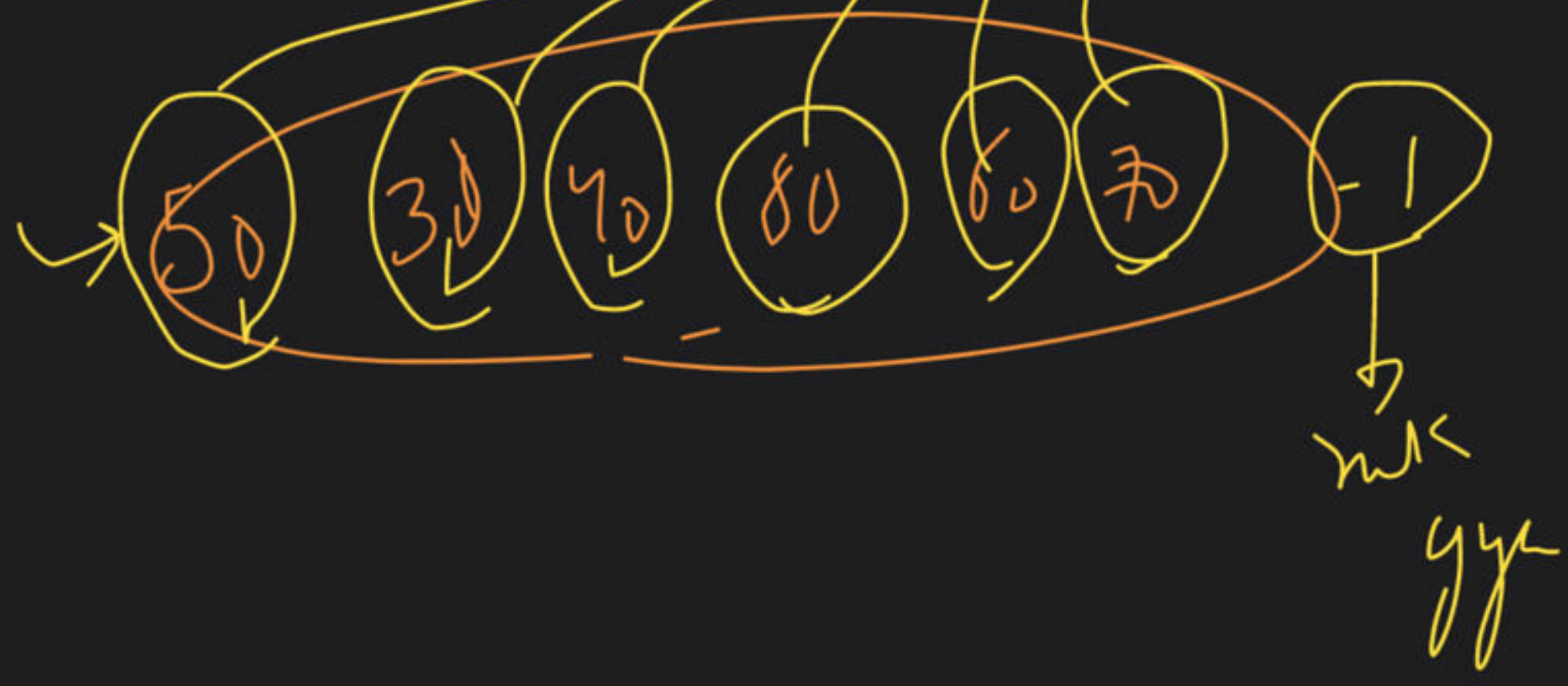
Val

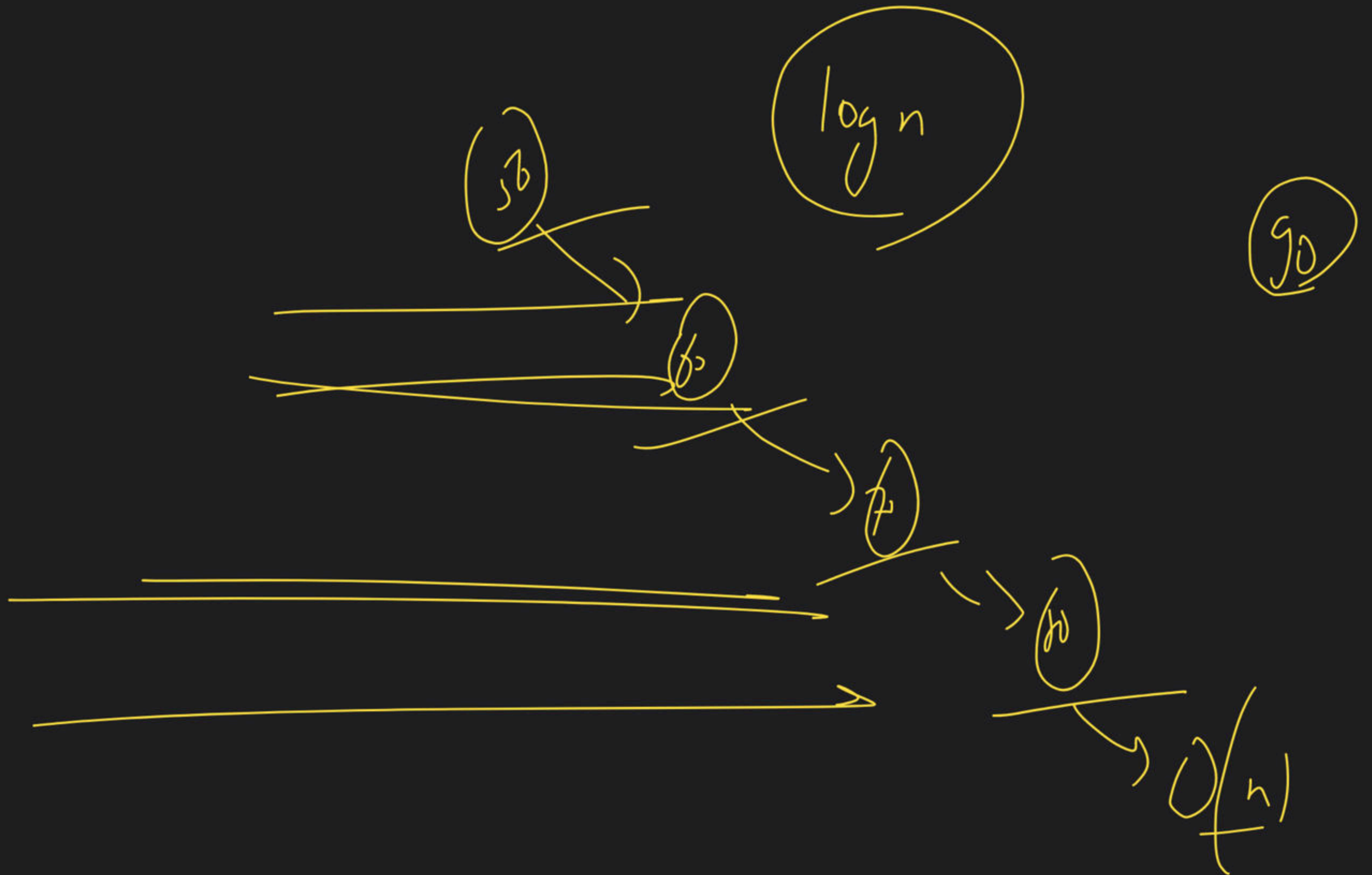
min

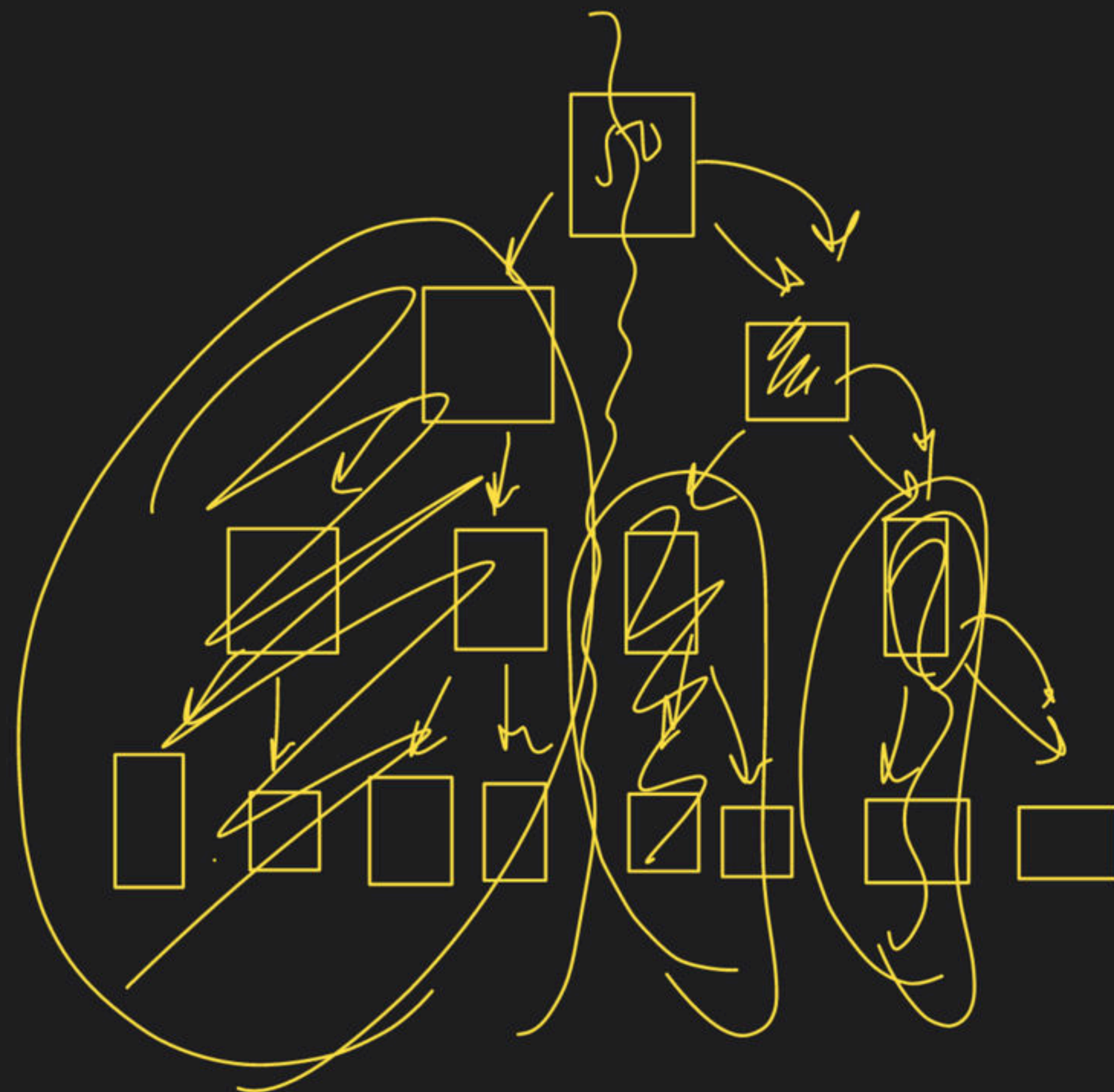




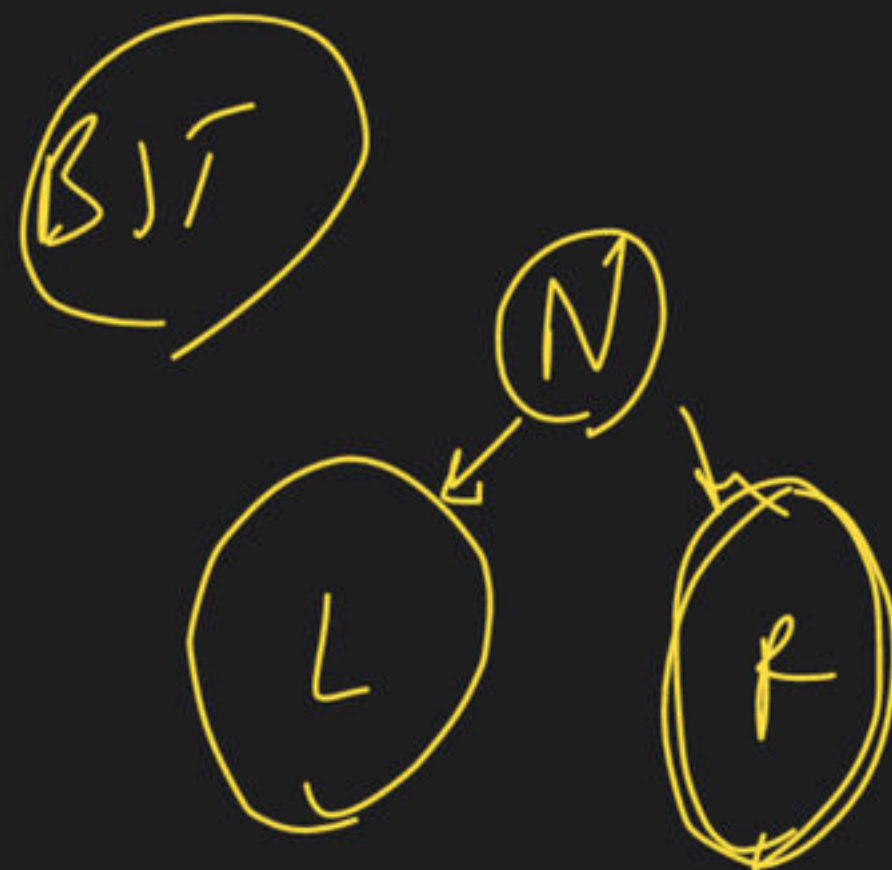
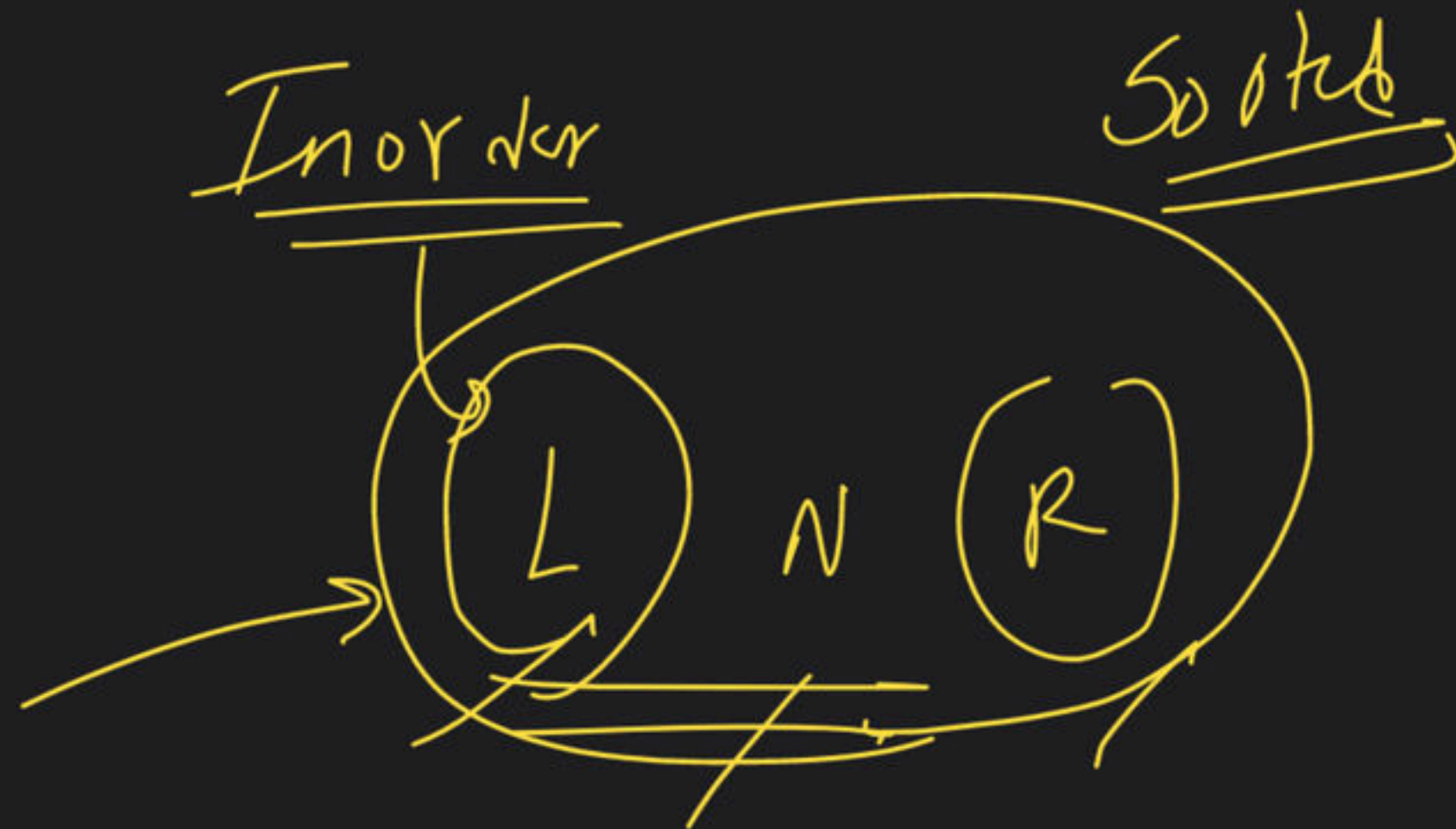
single node → tree me add krke







$n/2$
 $n/2$
 $n/2$



$$\begin{aligned} N &> L \\ N &< R \end{aligned}$$

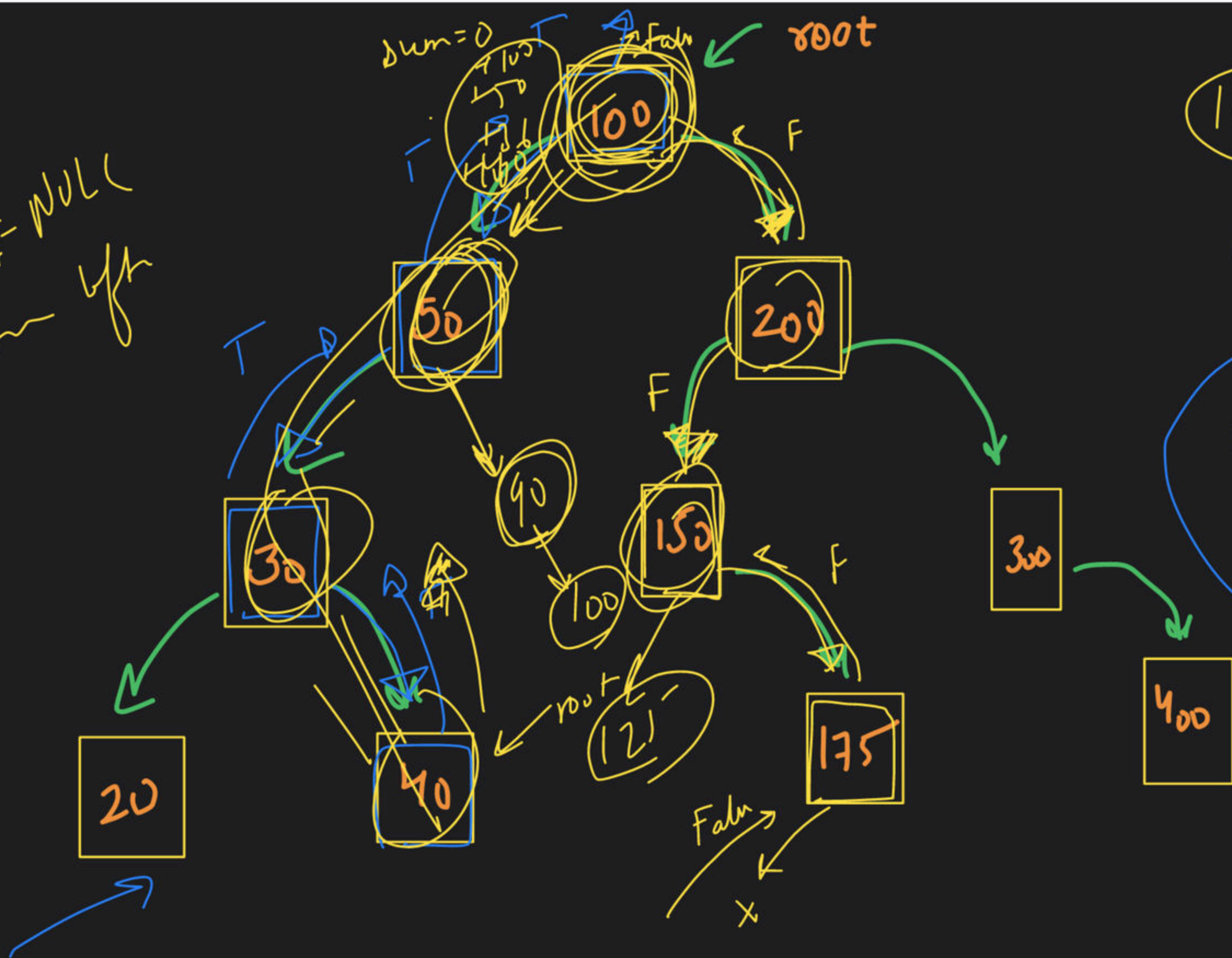
$$R > N > L$$

asc. order

NOTE

inorder of a
BST is
always
SORTED

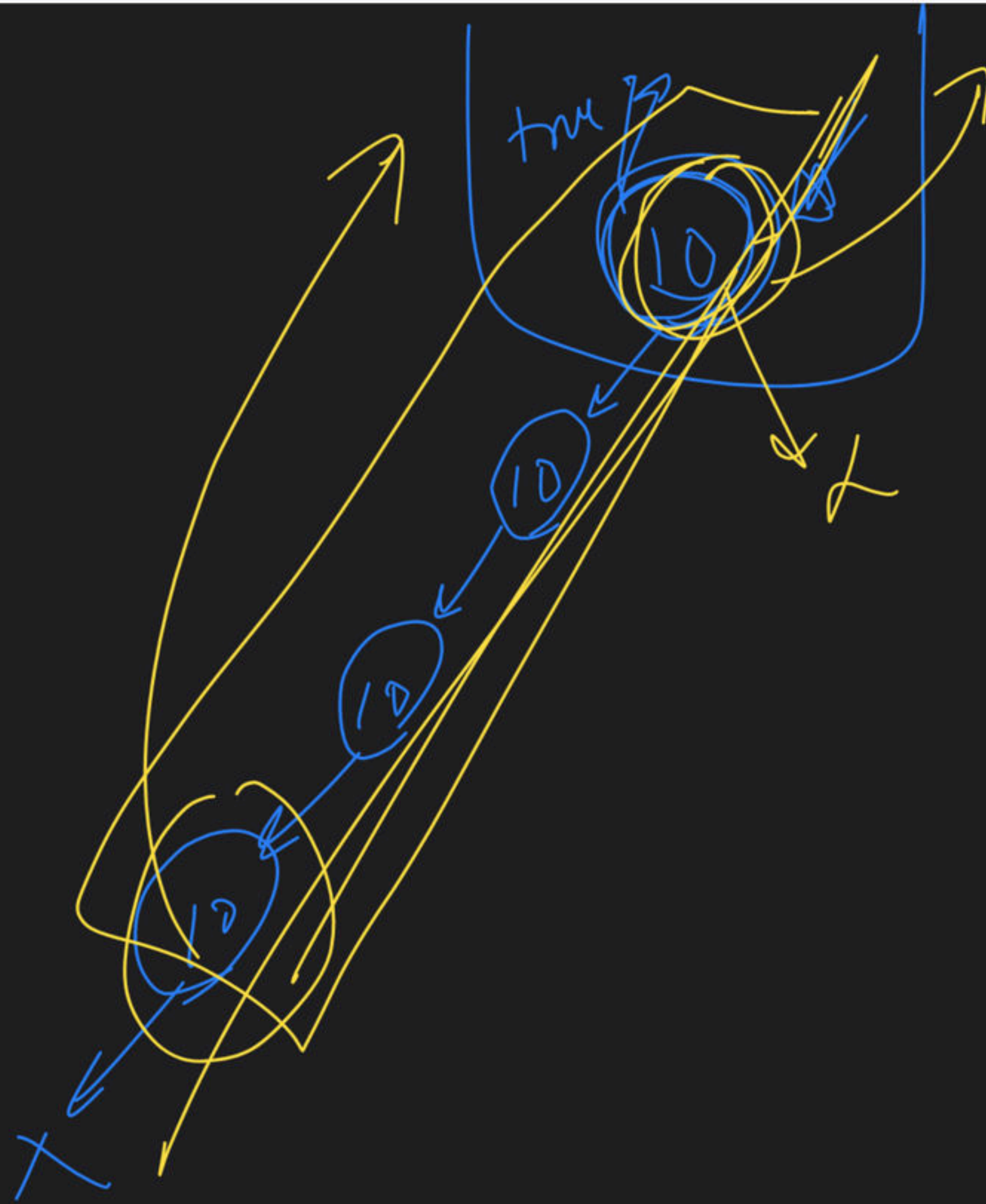
if left = null
return yfr



160 \Rightarrow target

$u_v \Rightarrow \text{target}$

$$\text{while}_c((t)!) = (\underline{-1})$$



10 10 10 10

$$\pi = 10$$

return left || right

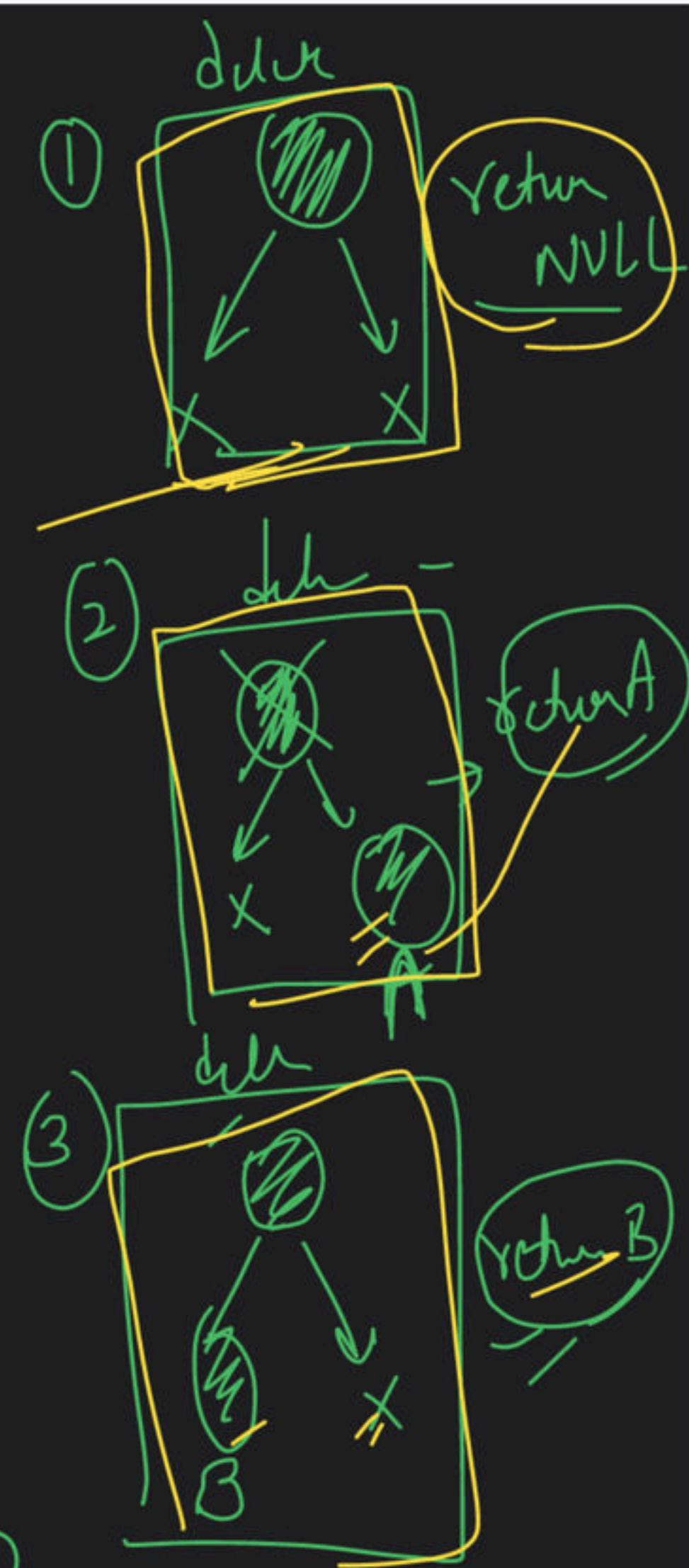
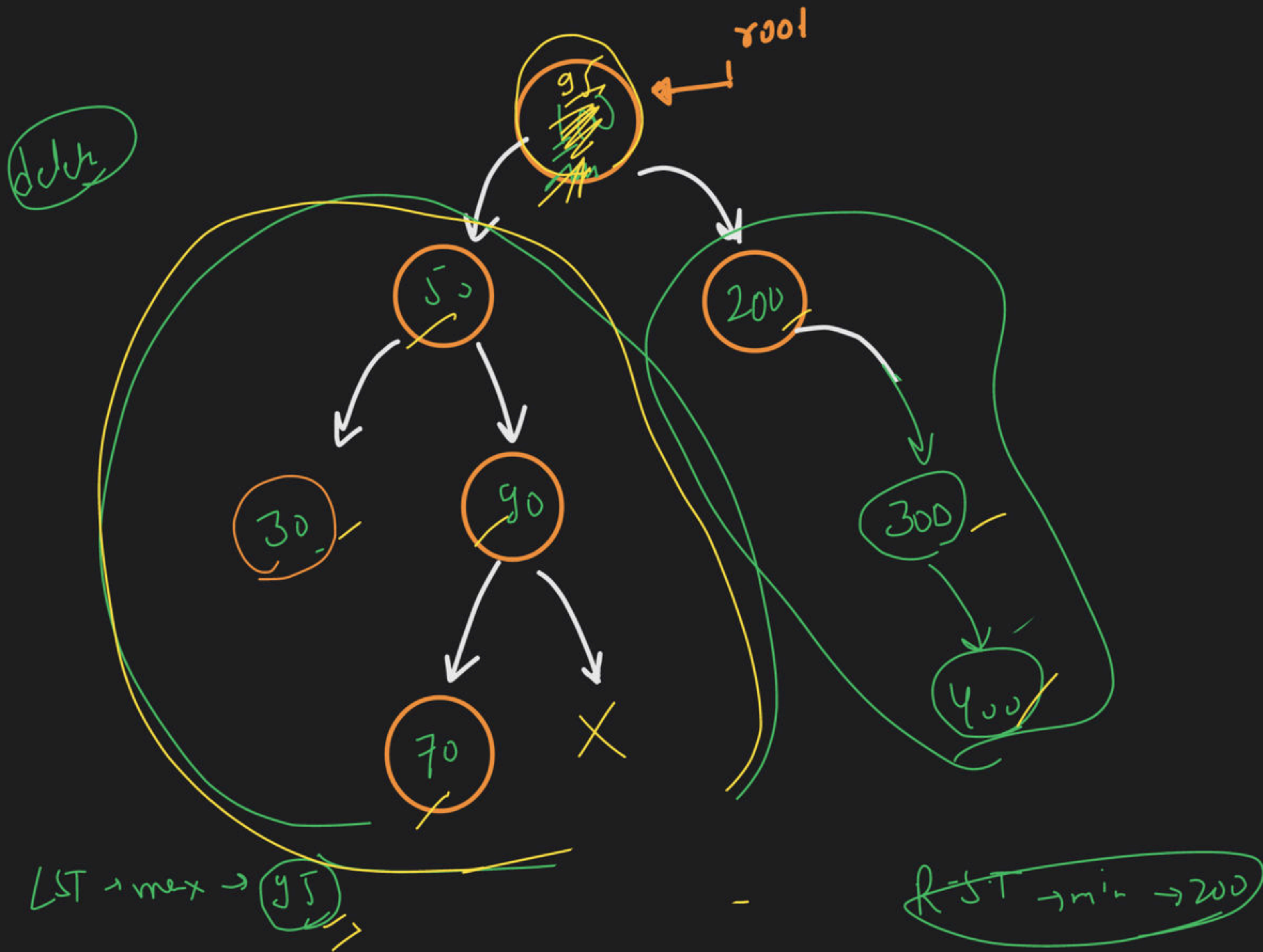
if (left == true)
return true

if (right == true)
return true

if (left == false & right == false)
return false

→ Delete a Node from BST





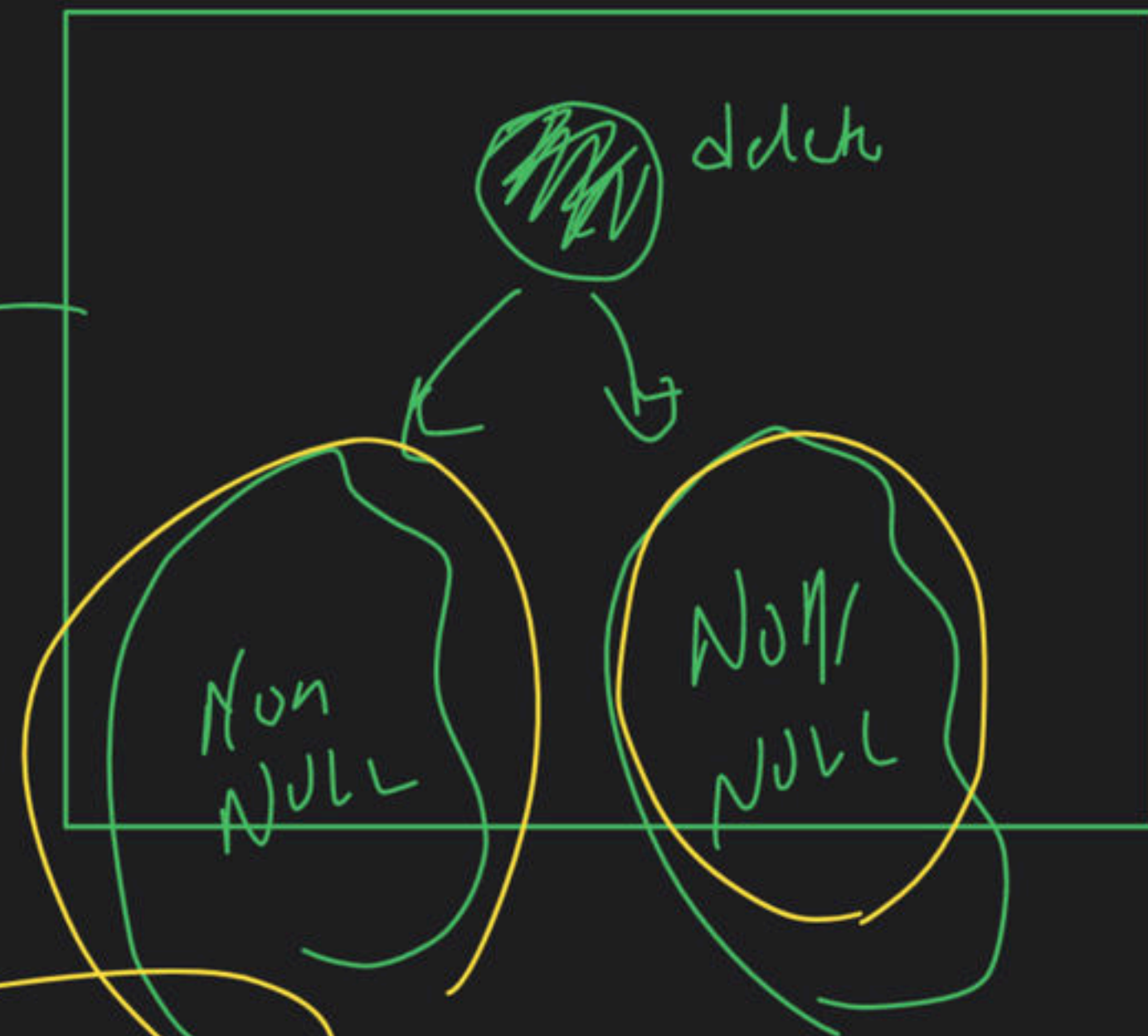
(I) L.S.T → max

Replace val

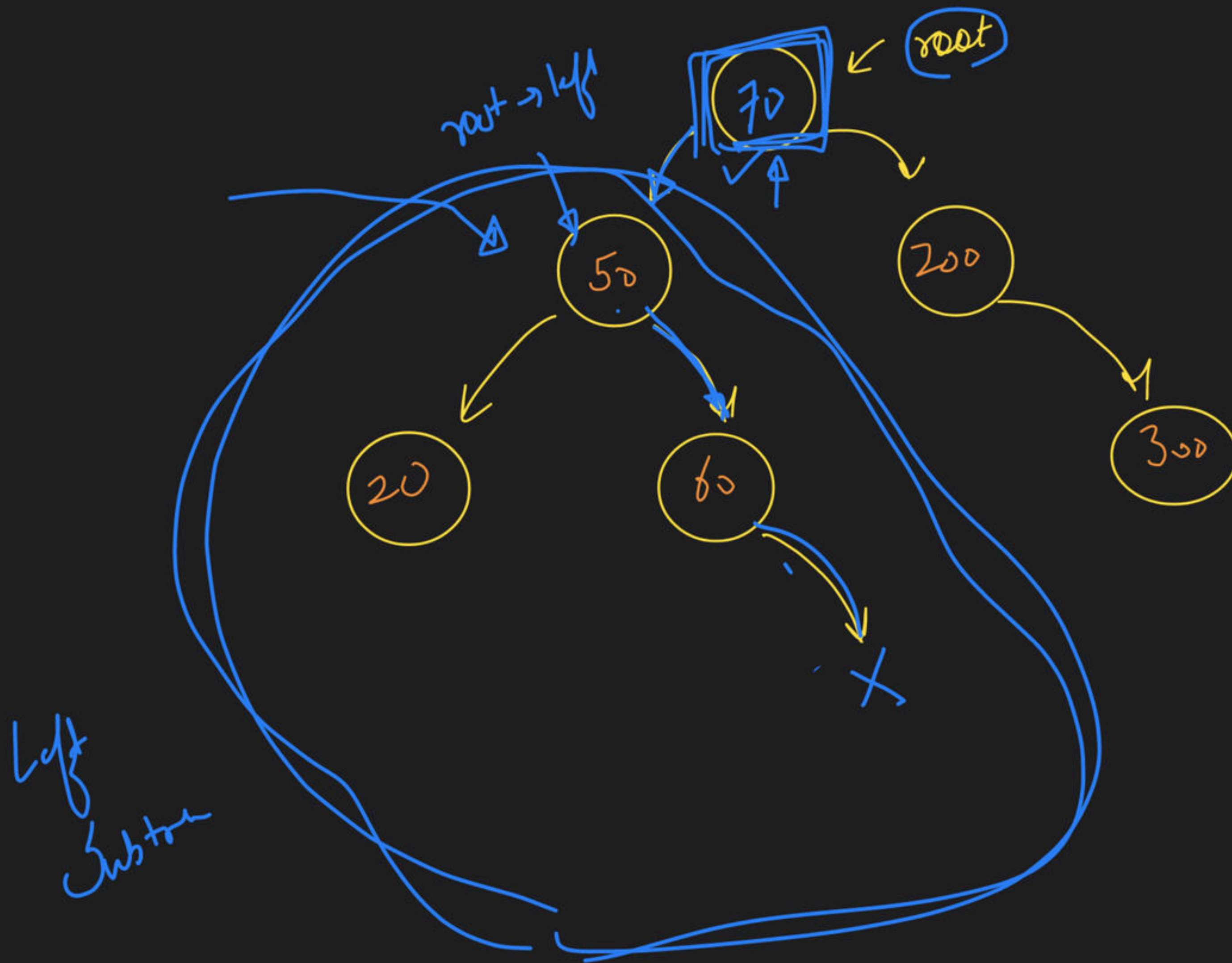
(II) R.S.T → min Val

(2)

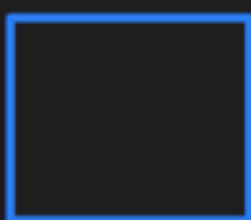
delete from left (90)



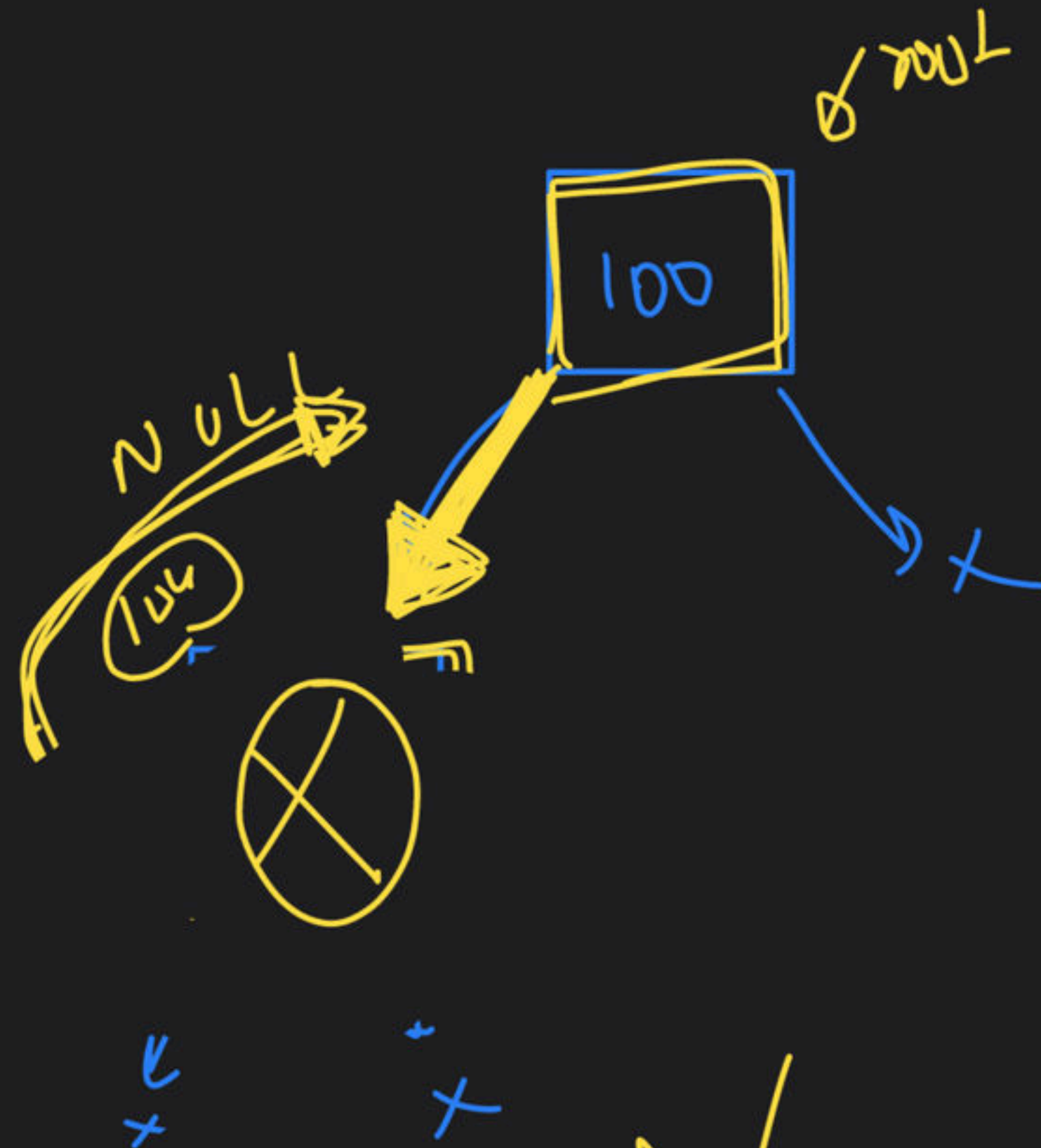
target = 100



- (1) 70 → null
- (2) 50 < 100 < 70
- (3) LST
→ 70
→ null



del.

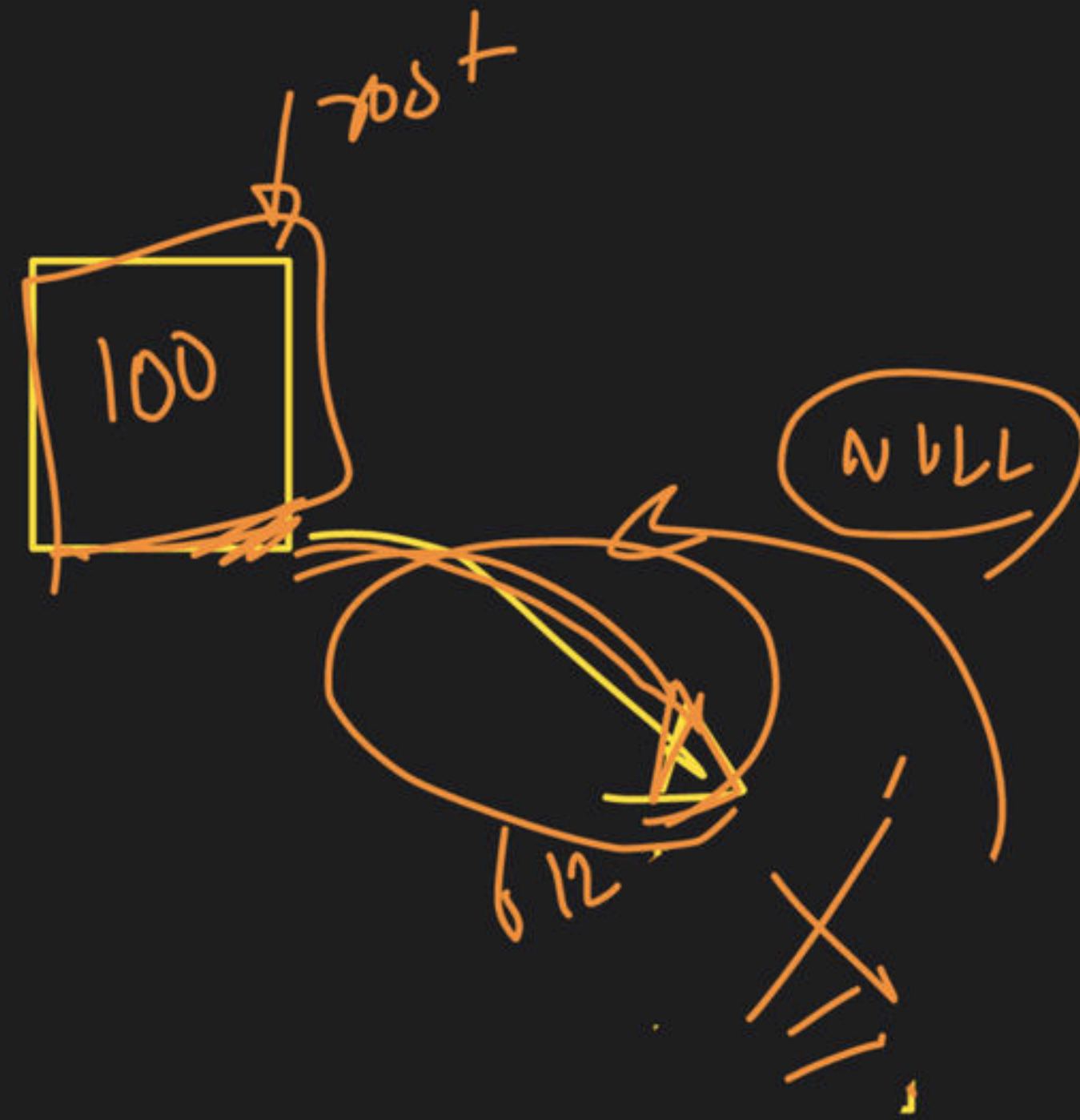


50 → target

~~root → left~~

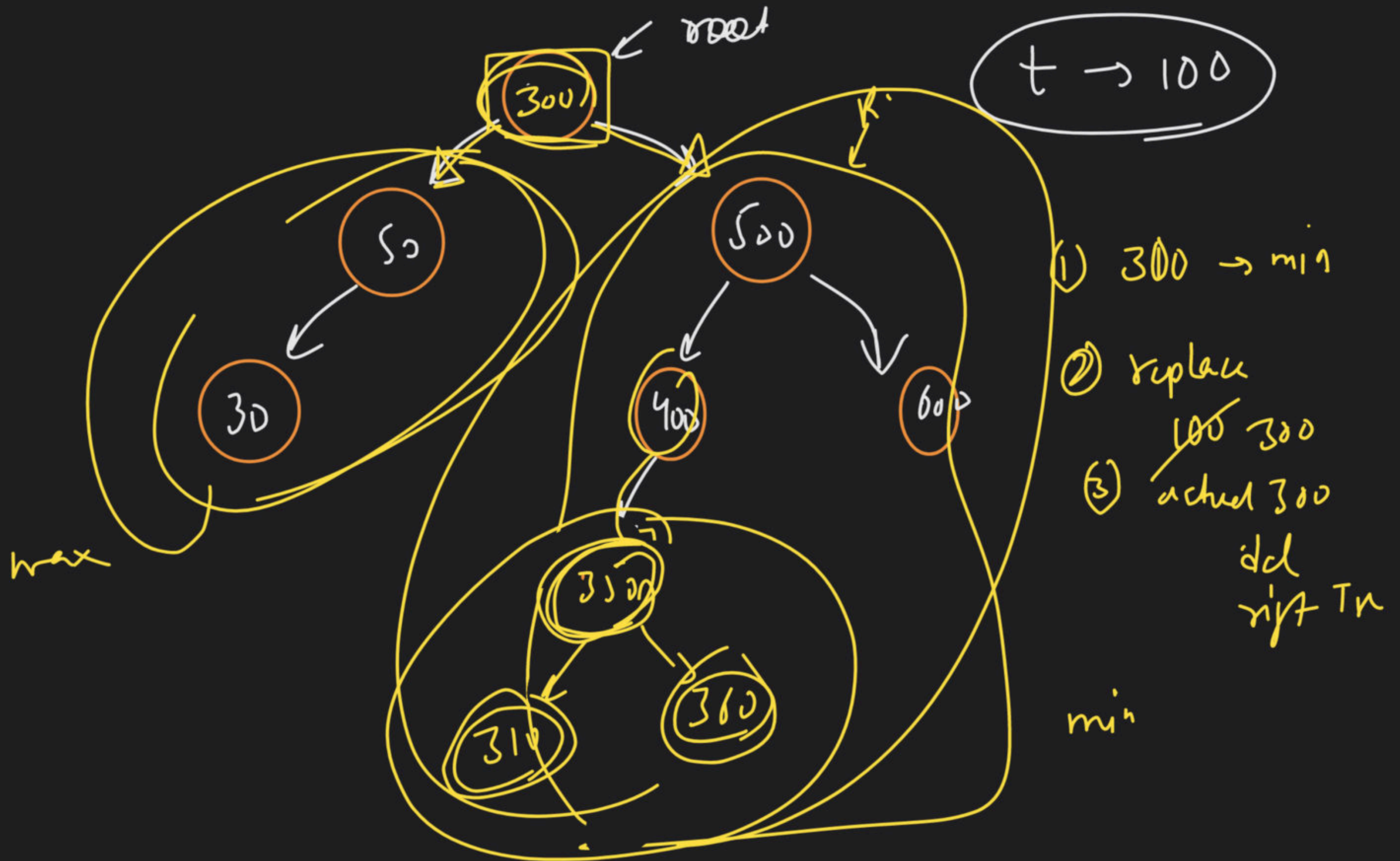
root → w, = L

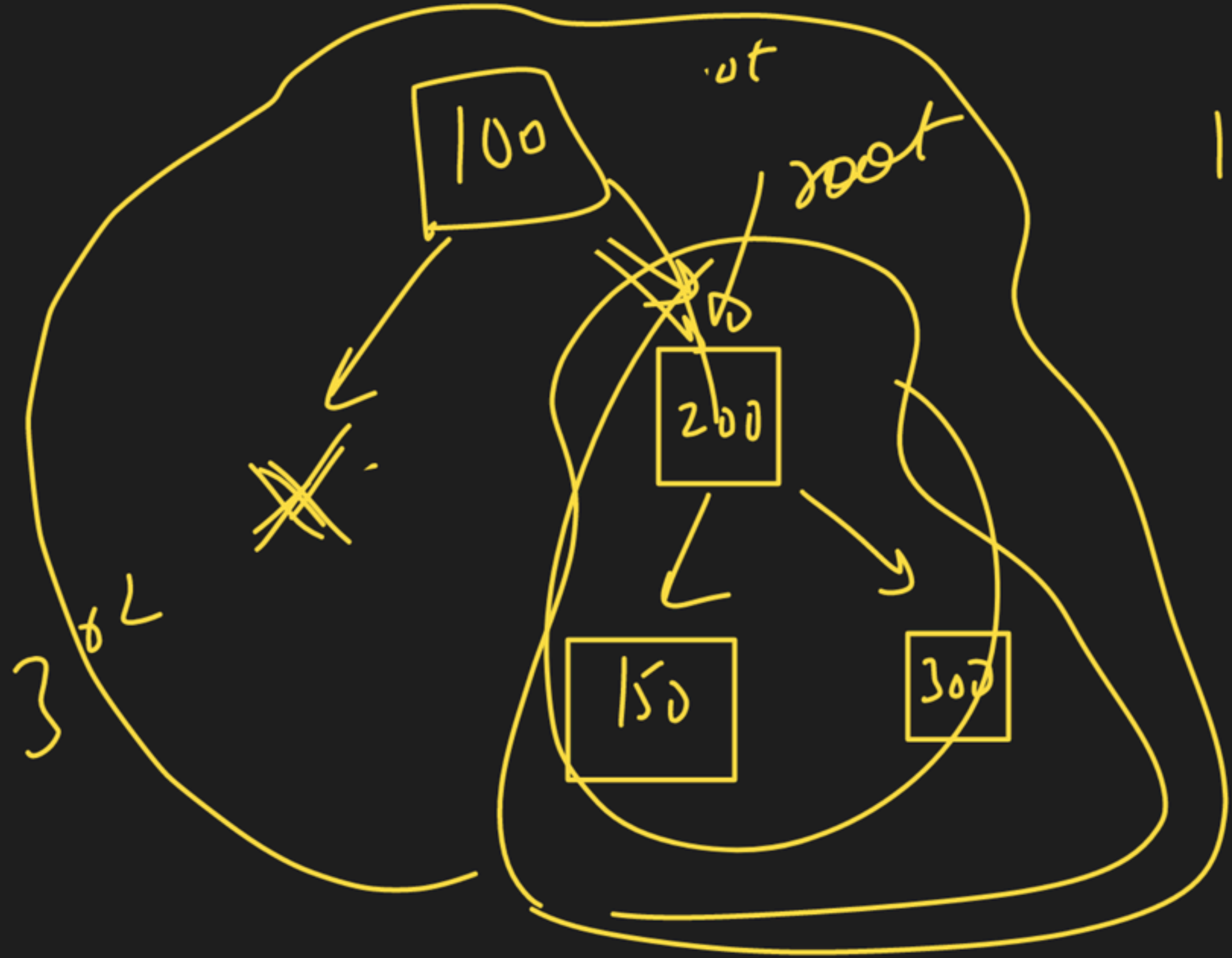
~~del~~



$t \rightarrow 200$

$root \rightarrow right = NULL$





100 → target

















