

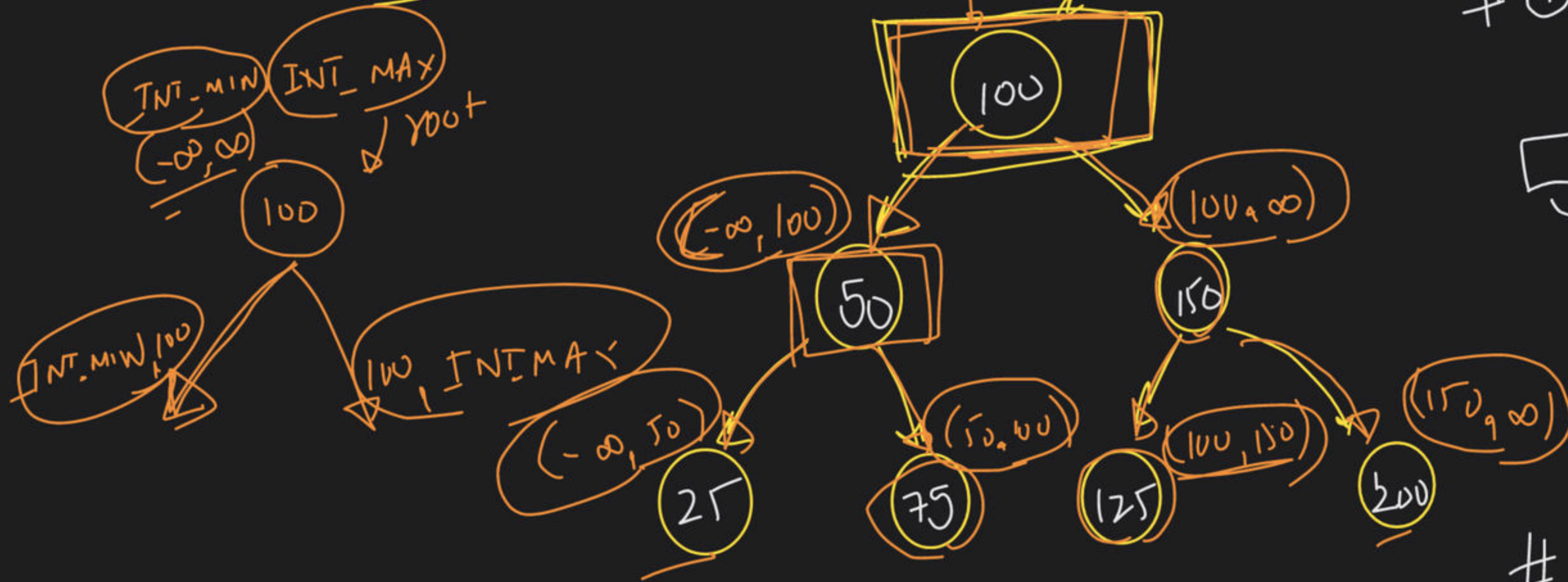
# BST Class - 2

Special class

→ ~~Tree~~

Height / Diameter of BST

# ① Validate BST



#① inorder  $\rightarrow O(n)$



#2  $\rightarrow$  Recursion

Valid BST







LCA

BST

Bonus

$p = 70$

$q = 160$

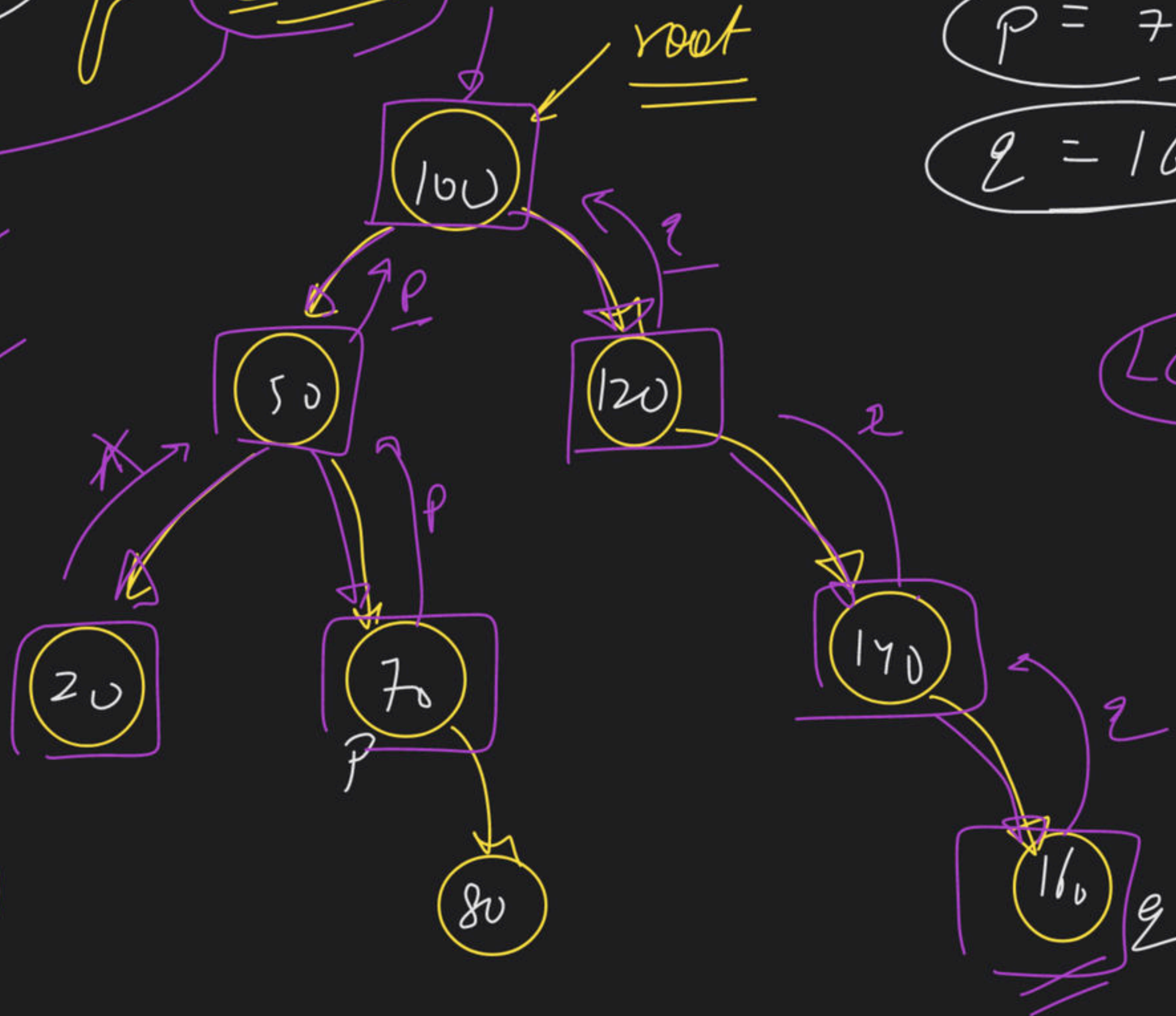
$LCA = 100$

(\*) inorder sorted  
(\*) BST property

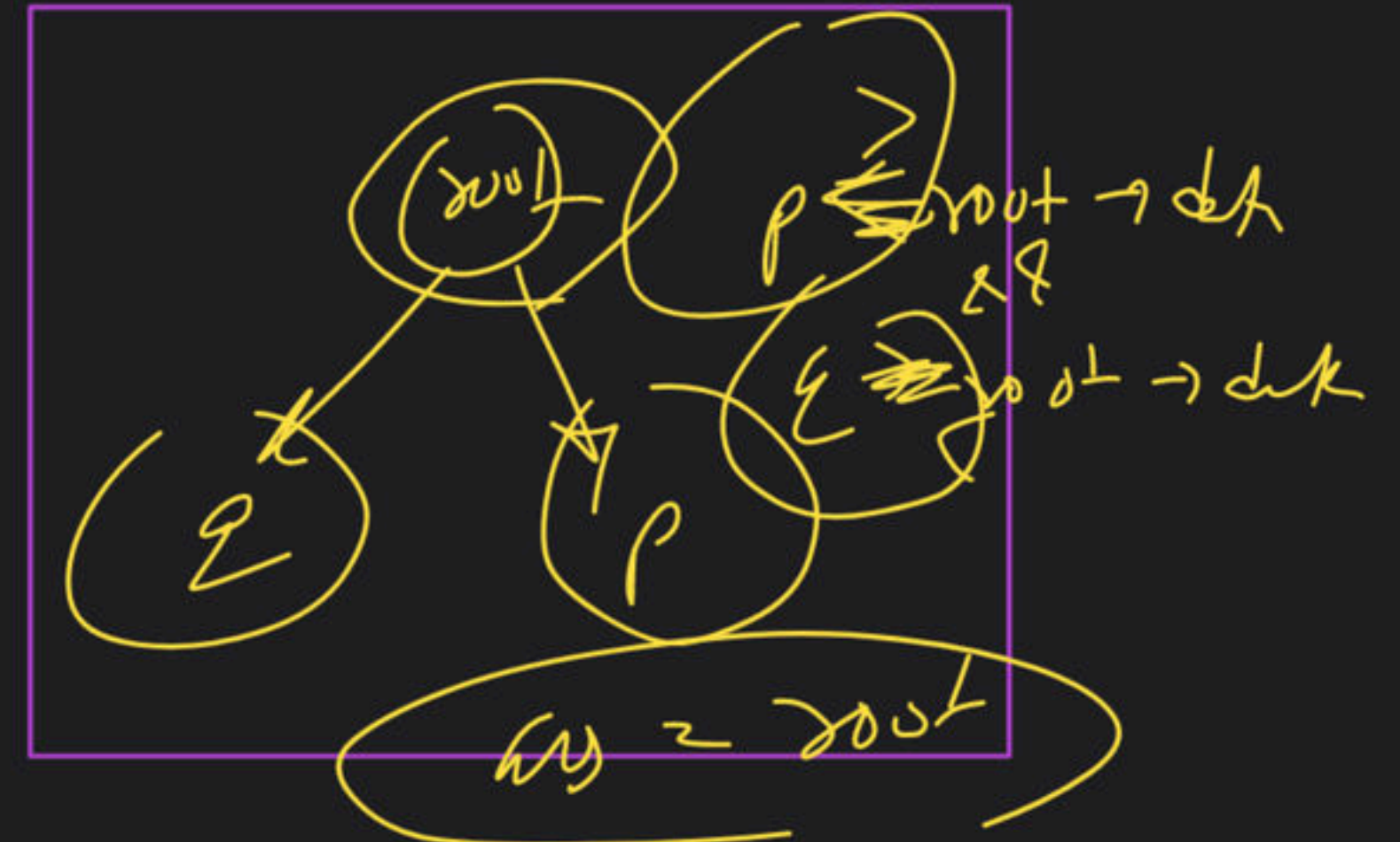
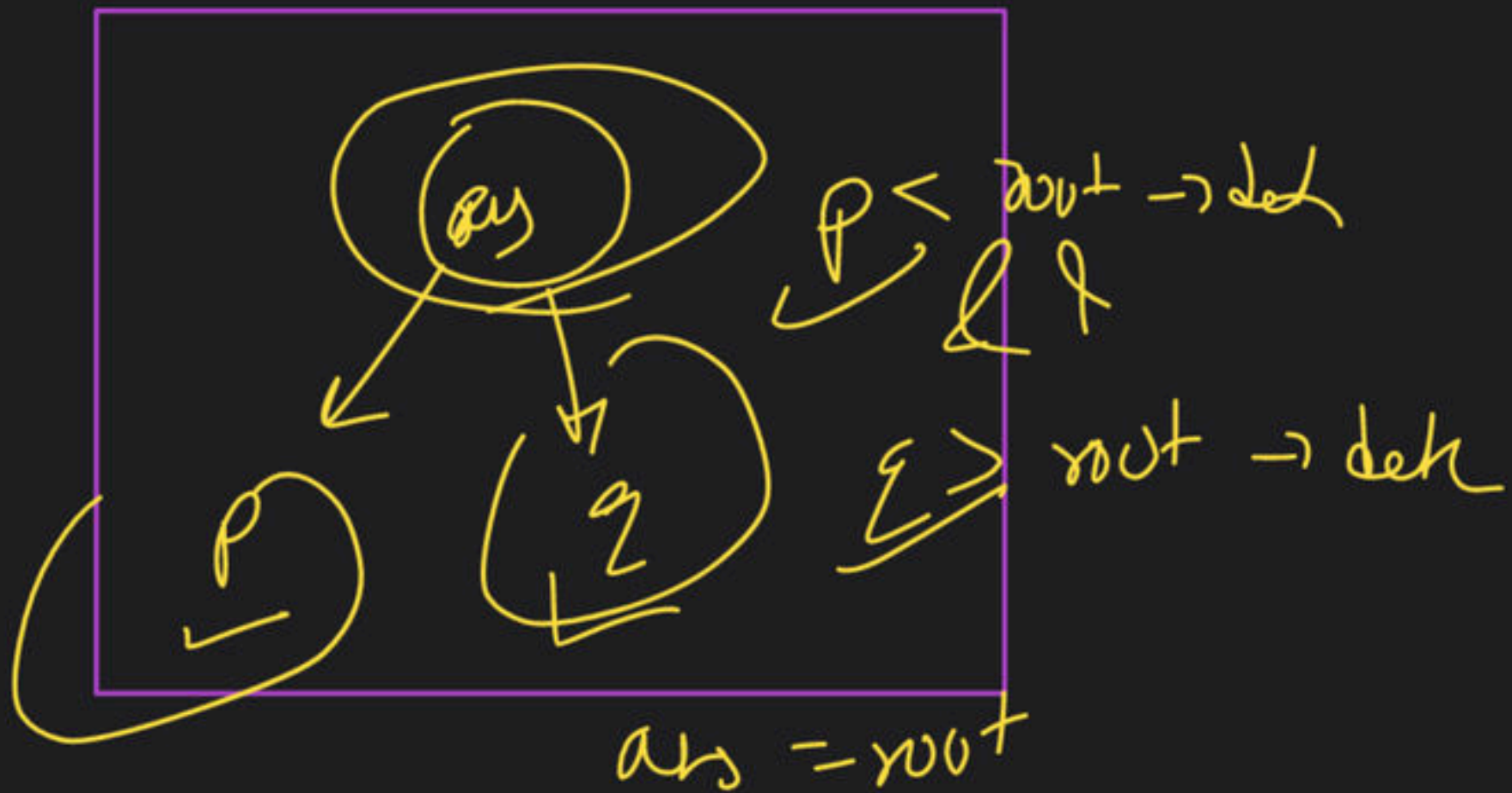
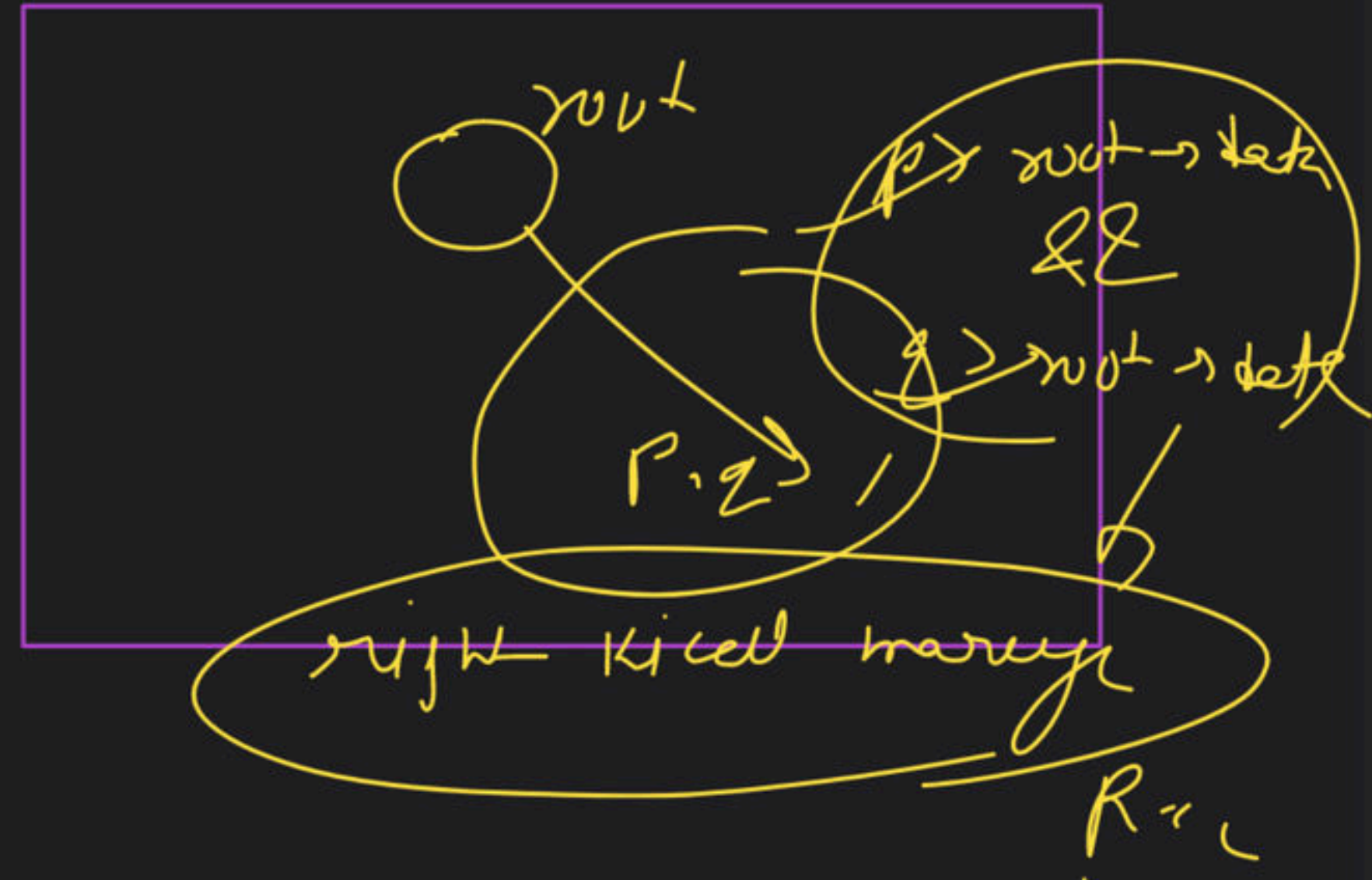
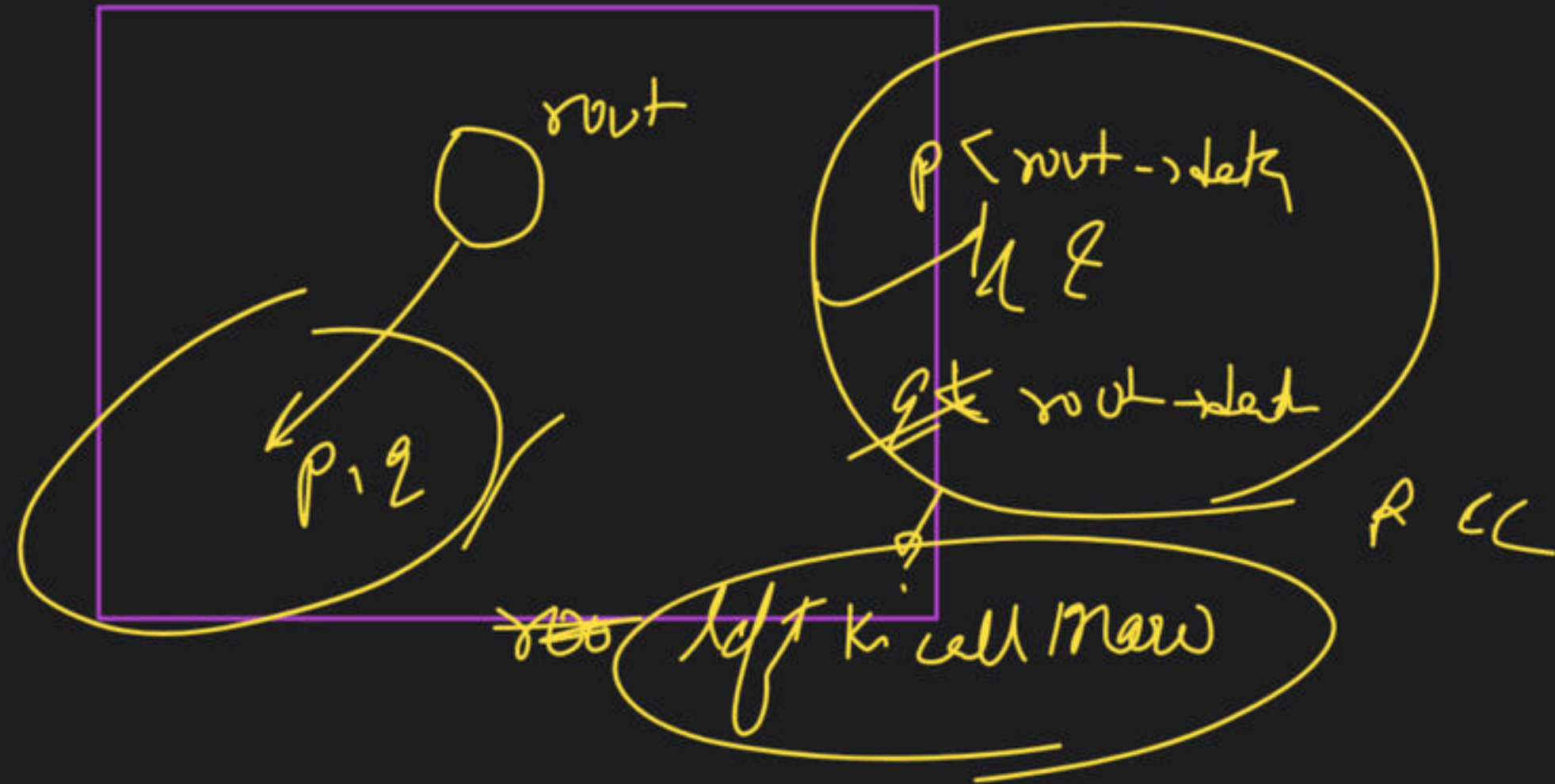
root

left

right





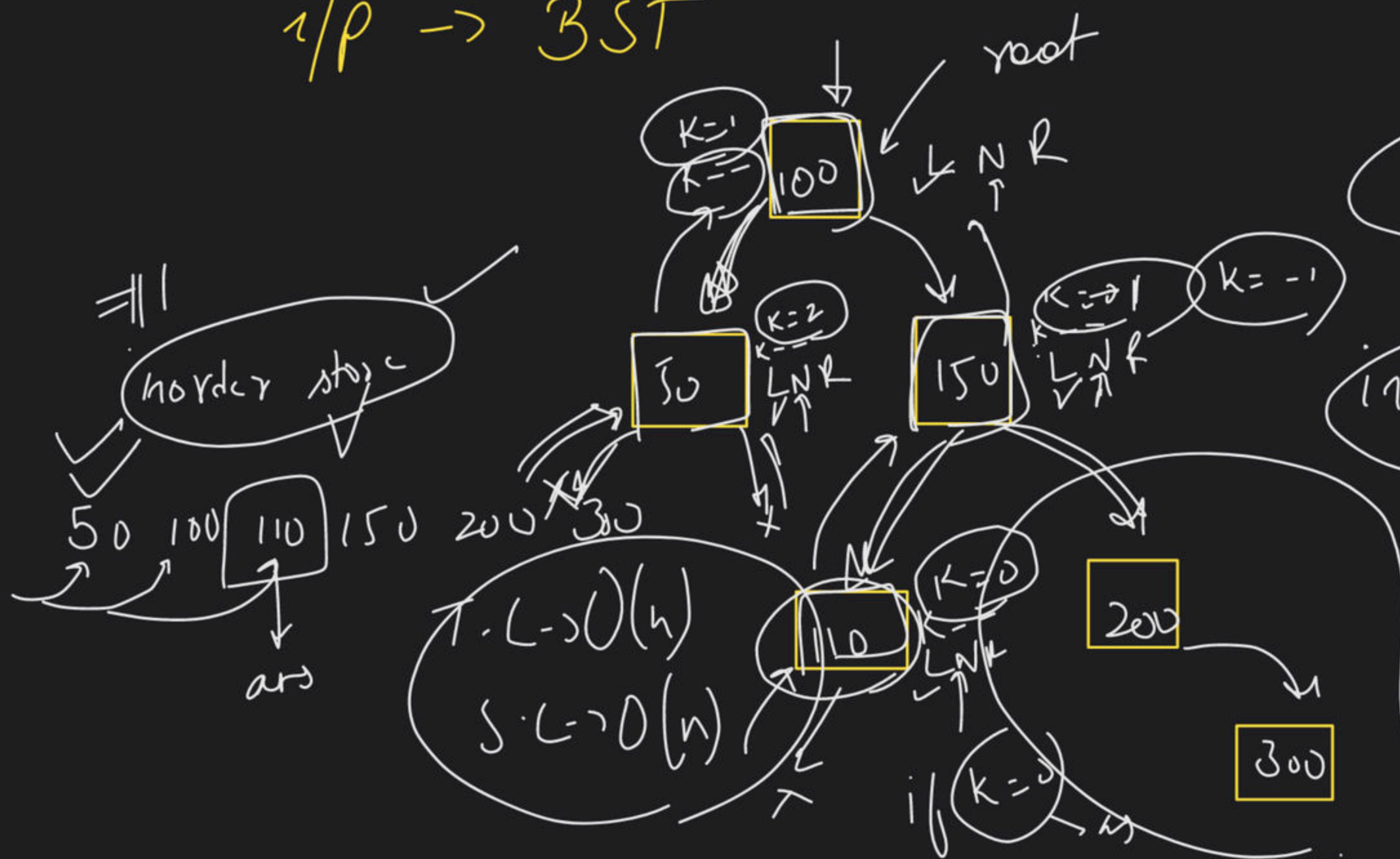




→  $K^{th}$  smallest element →

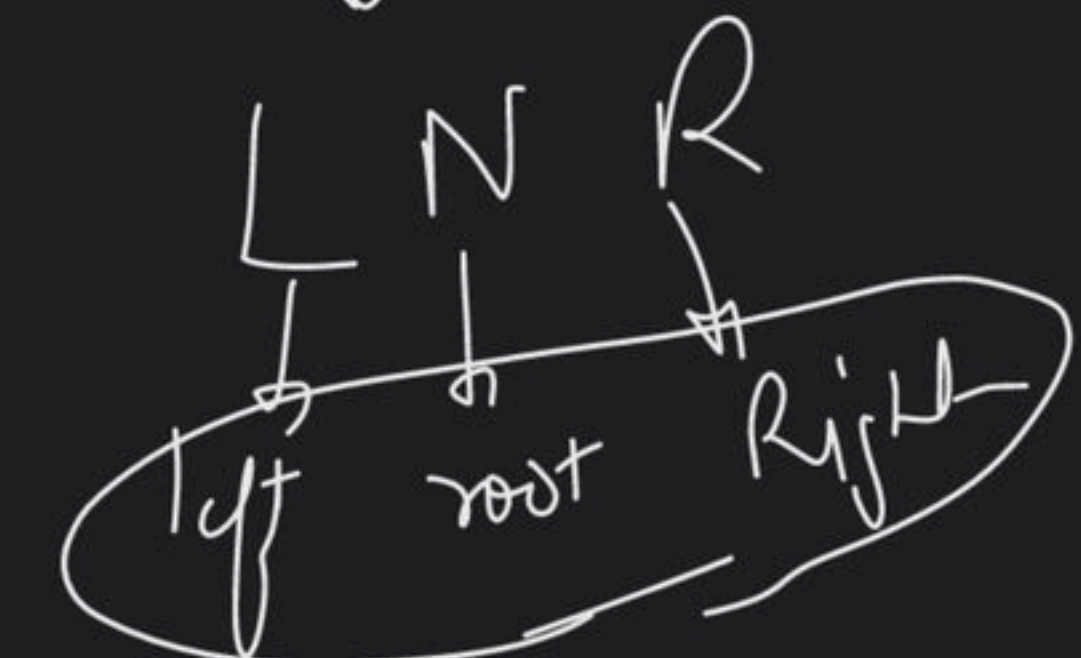
How? lmin

i/p → BST

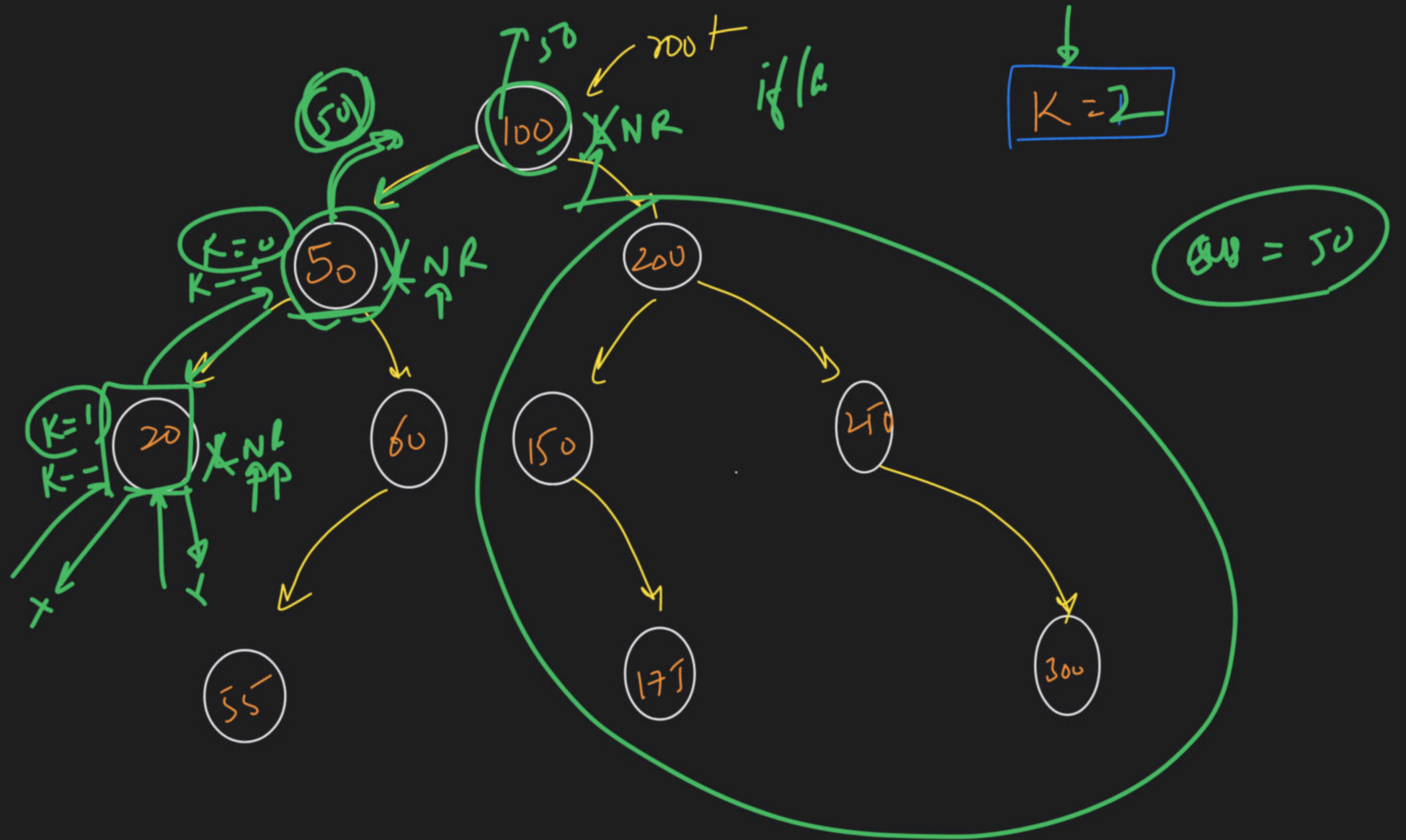


3<sup>rd</sup> smallest → K=3

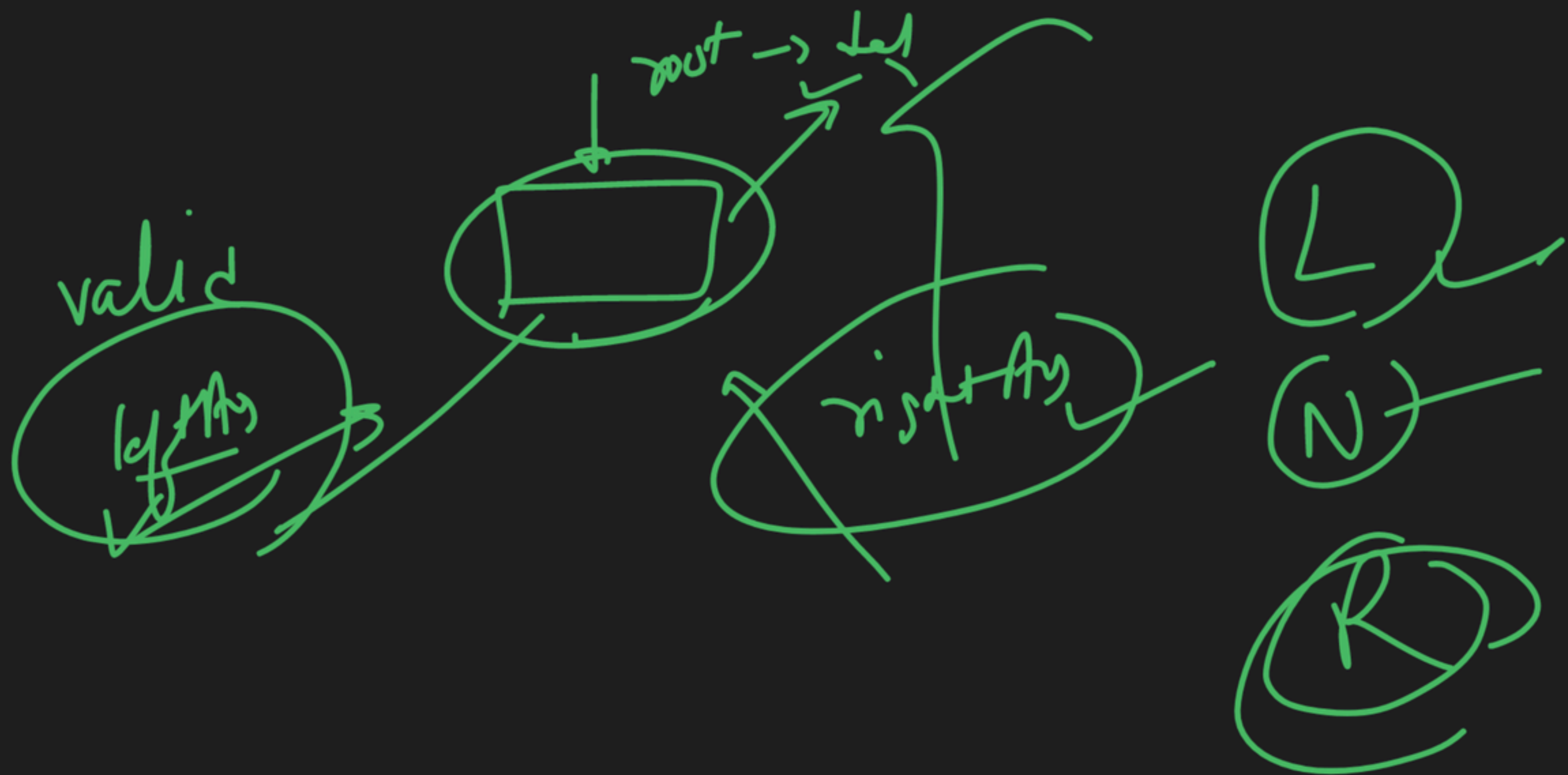
inorder traversal











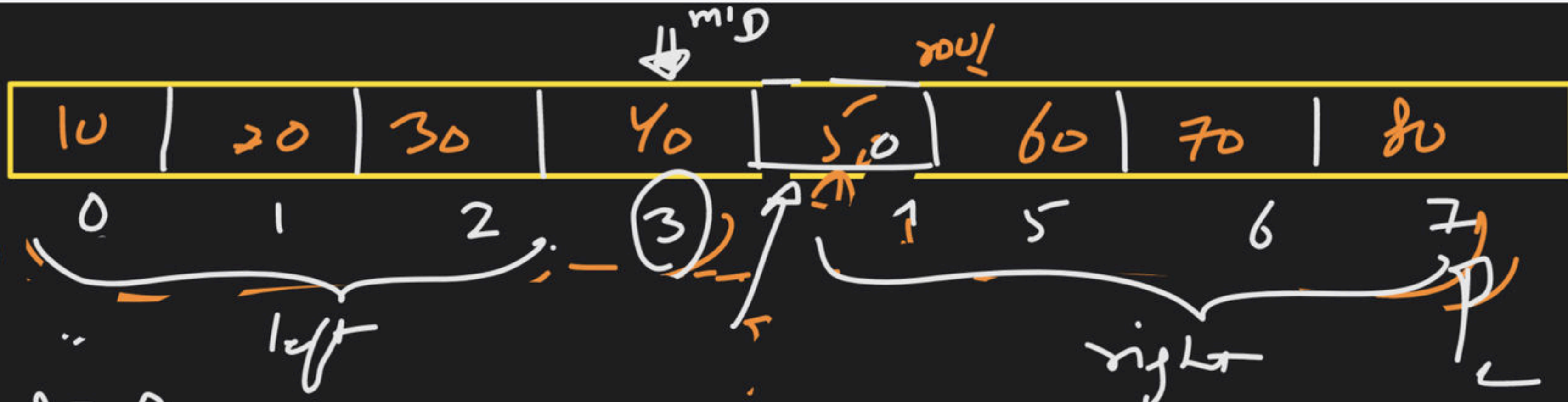


→ Inorder traversal → BST create

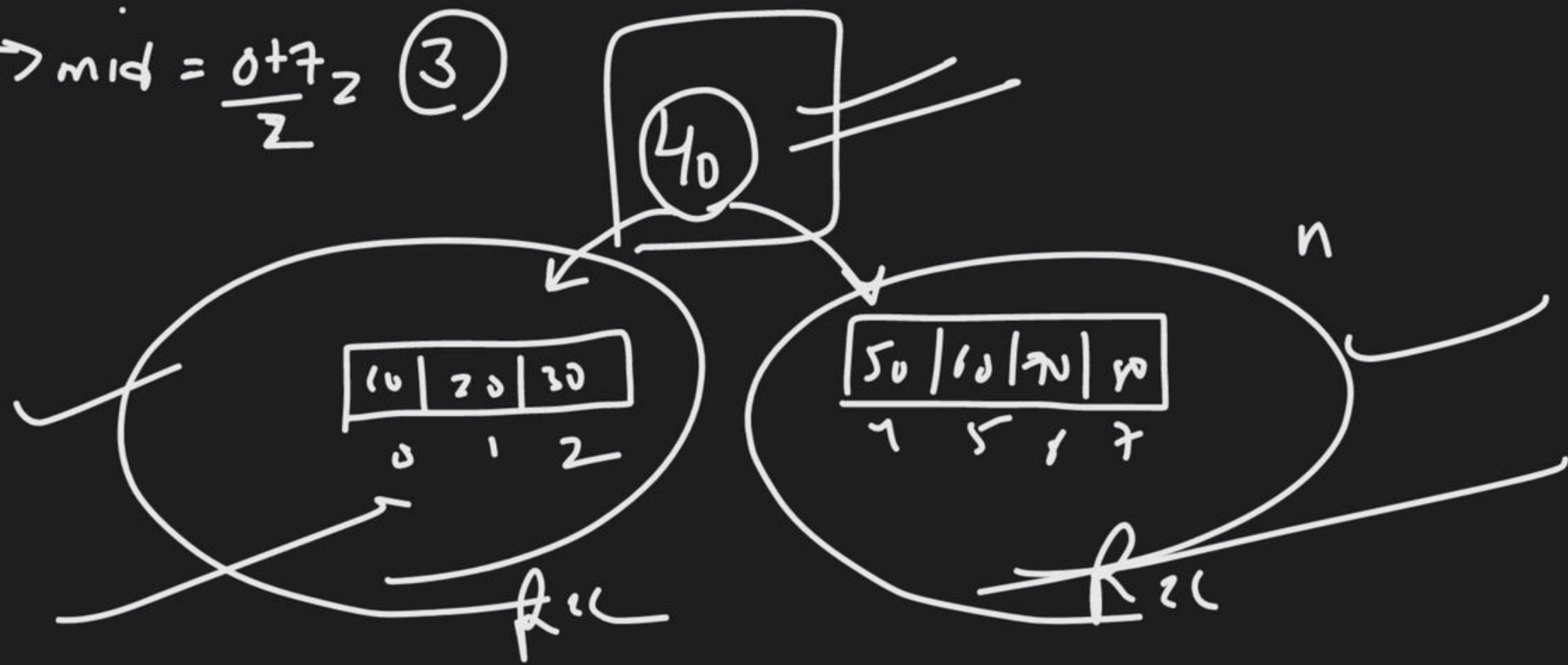
↓

sorted





$l = 0$   
 $r = 7$   
 $mid = \frac{0+7}{2} = 3$









1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

0 1 2 3 4 5 6 7 8

mid

$s=0$   
 $e=8$   
 $mid=4$

1	2	3	4
---	---	---	---

$s=0$   
 $e=3$   
 $mid=1$

5  
 2 7  
 1 3 8  
 4 9

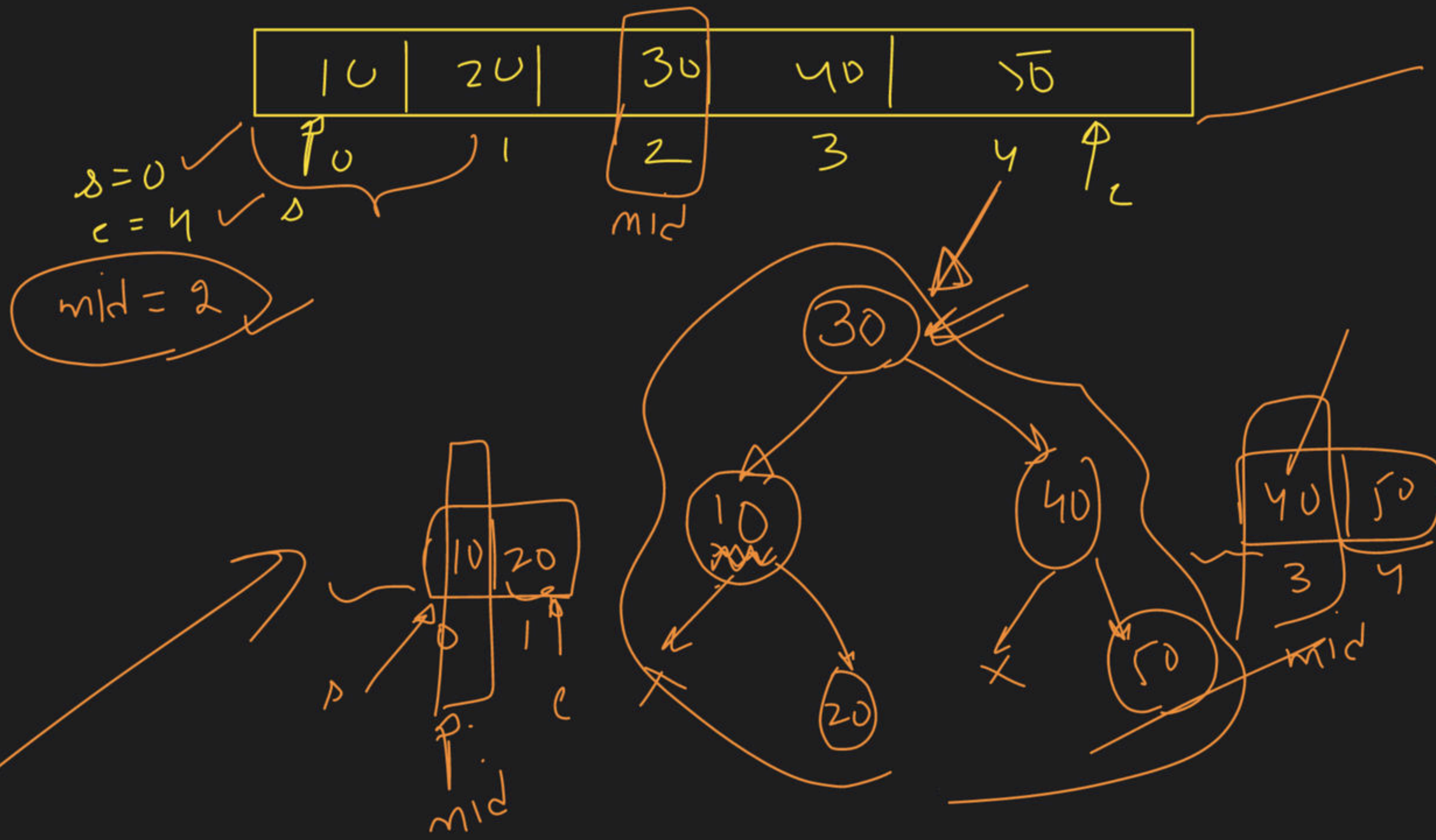


6	7	8	9
---	---	---	---

$s=5$   
 $e=8$

not a tree  
 why







→ Convert a BST into a Balanced  
BST

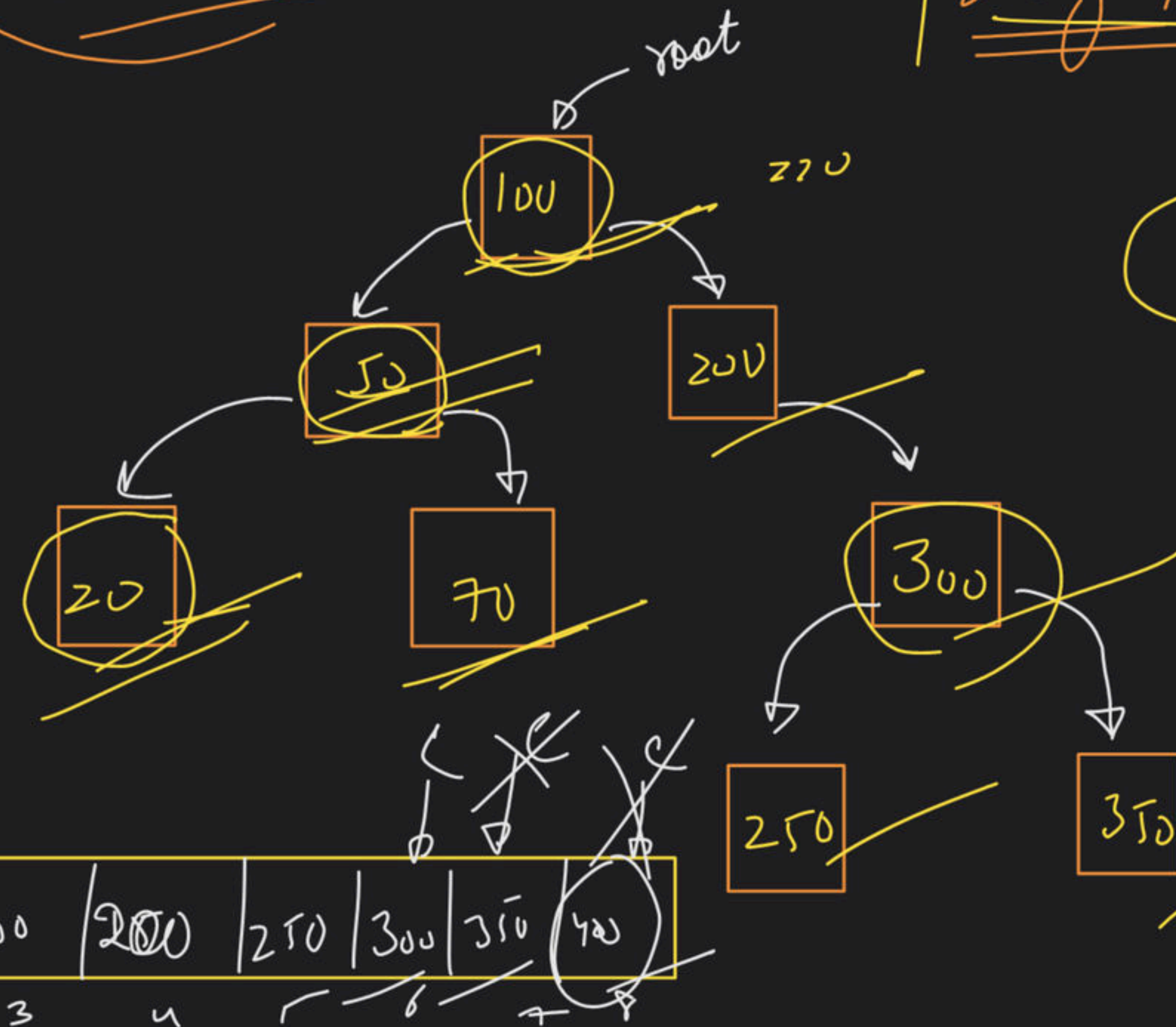


2 - sum

target = 320

BST:-

sorted array



(#11)

1st Node  
→ n options

1 option  
→ search  
→ log n

$n \log n$   
 $n^2$

20	50	70	100	200	250	300	350	400
1	2	3	4	5	6	7	8	9















































