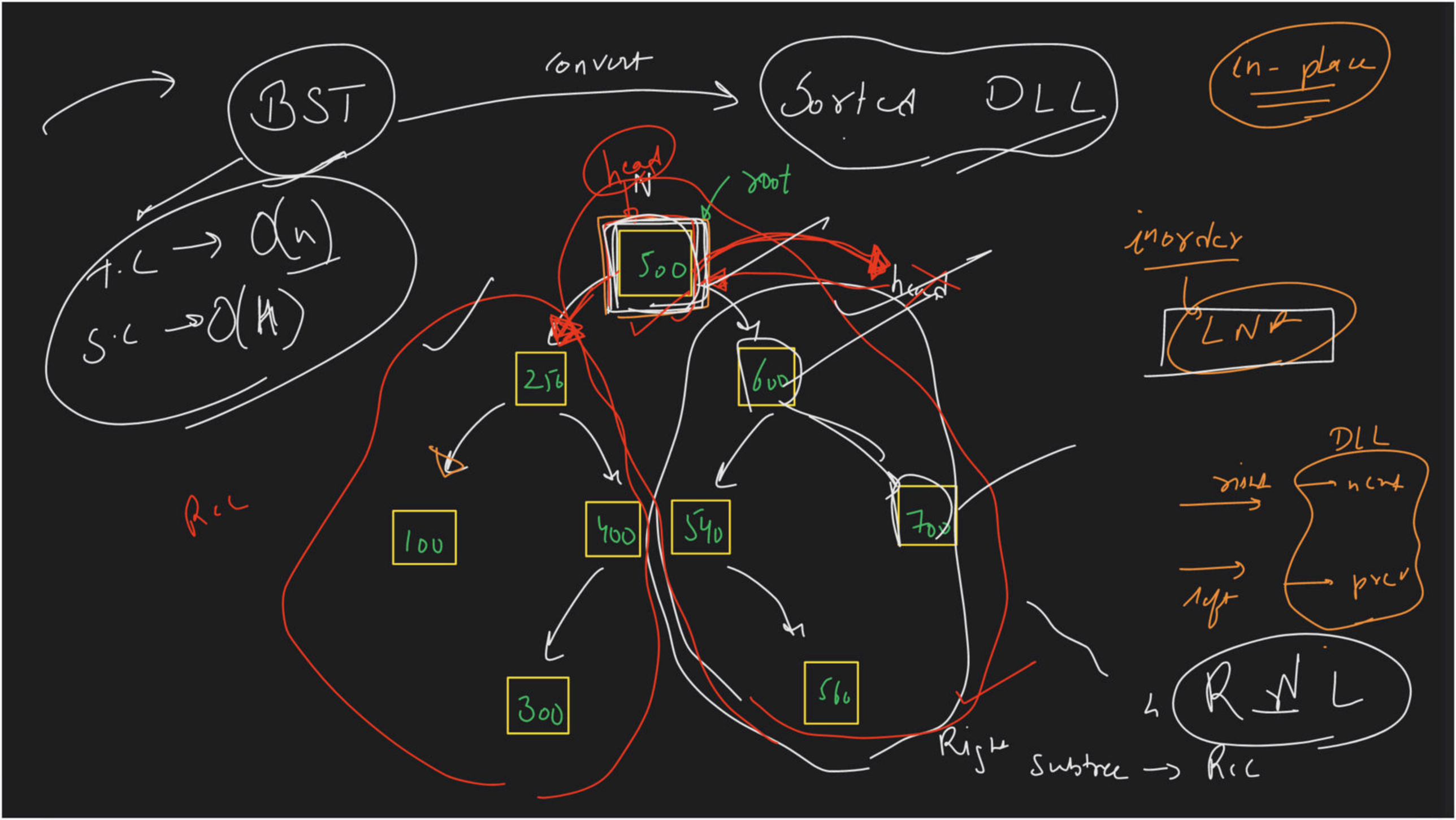
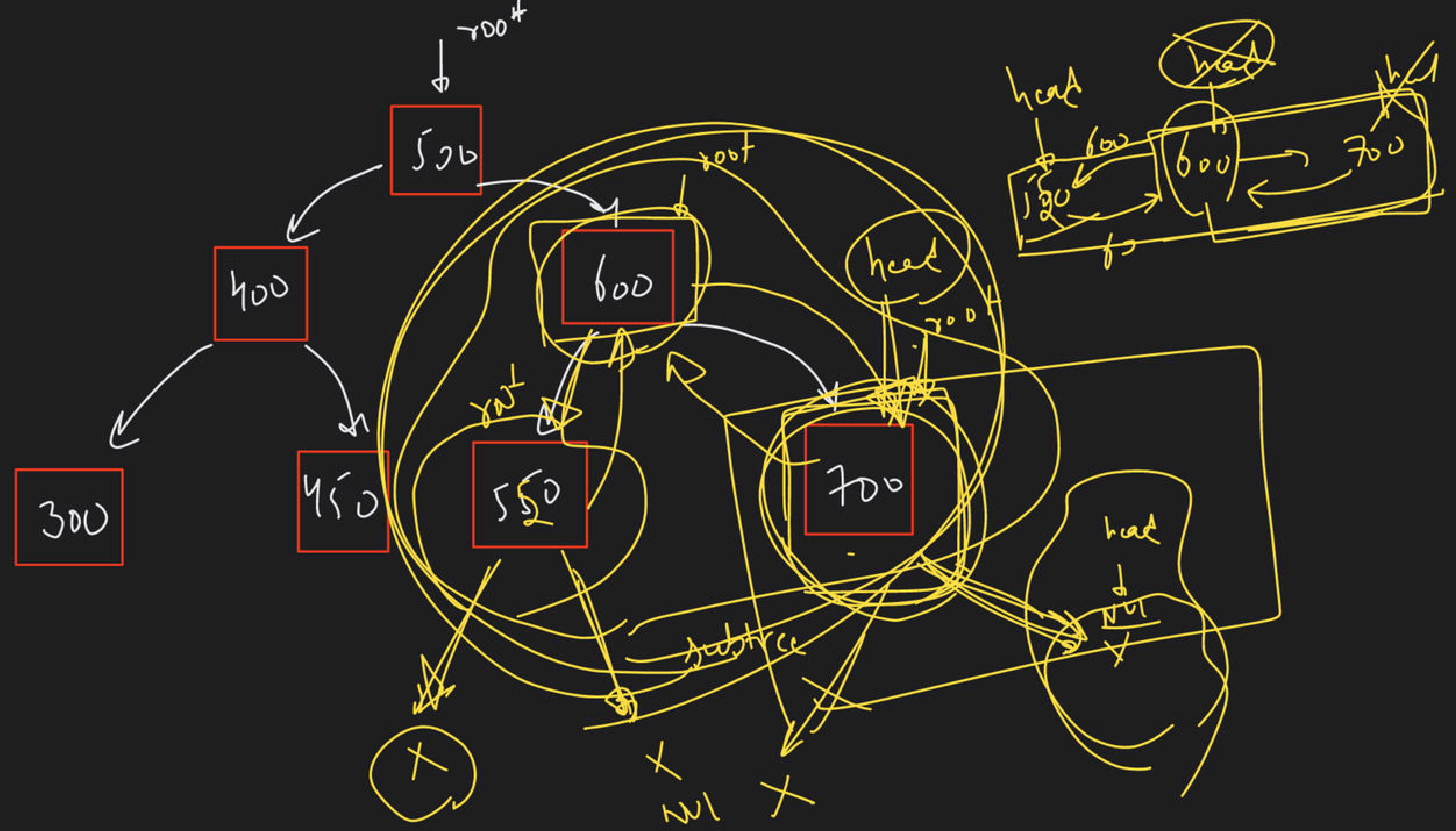
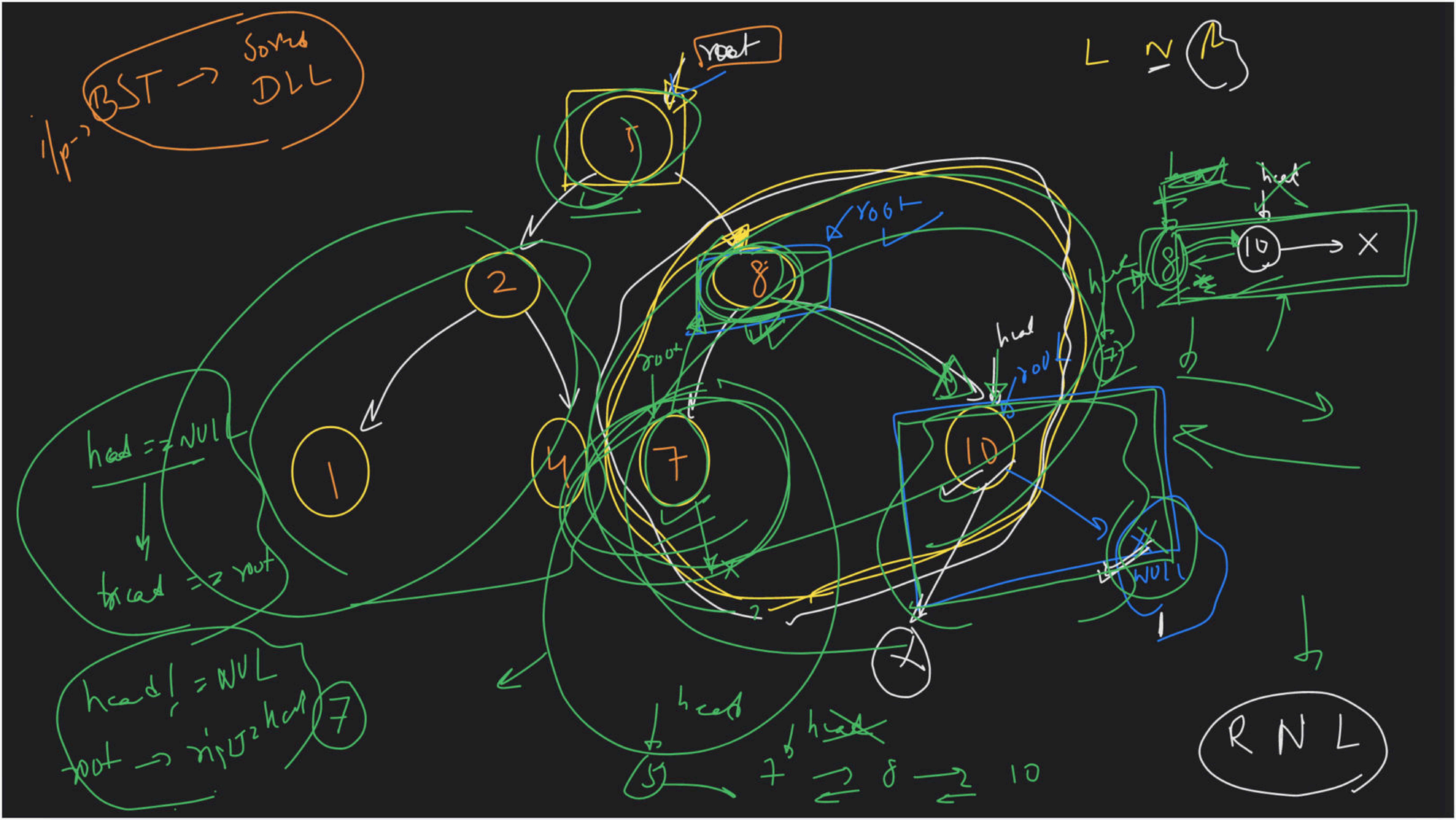
BST Class - 3

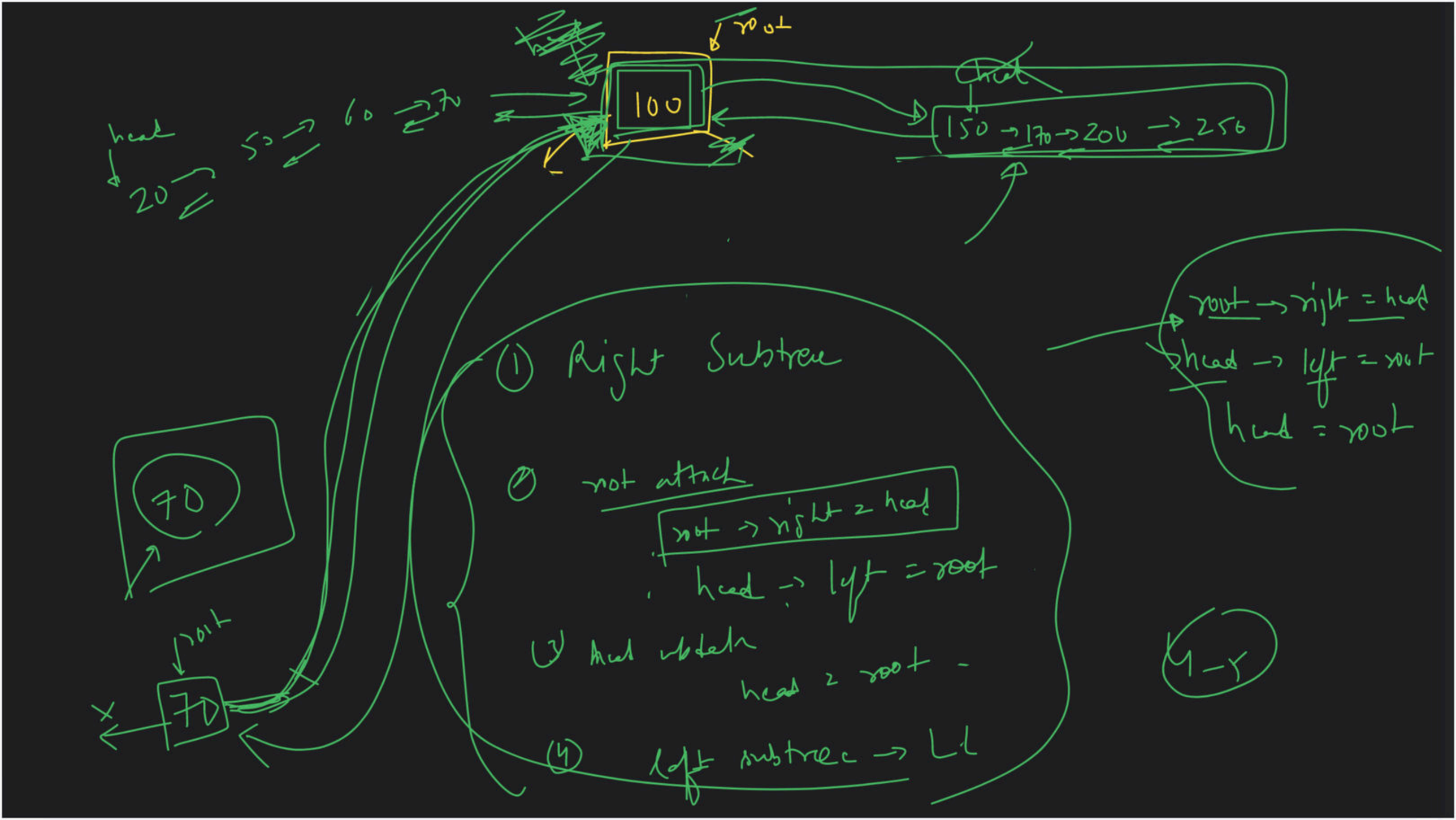
Special class

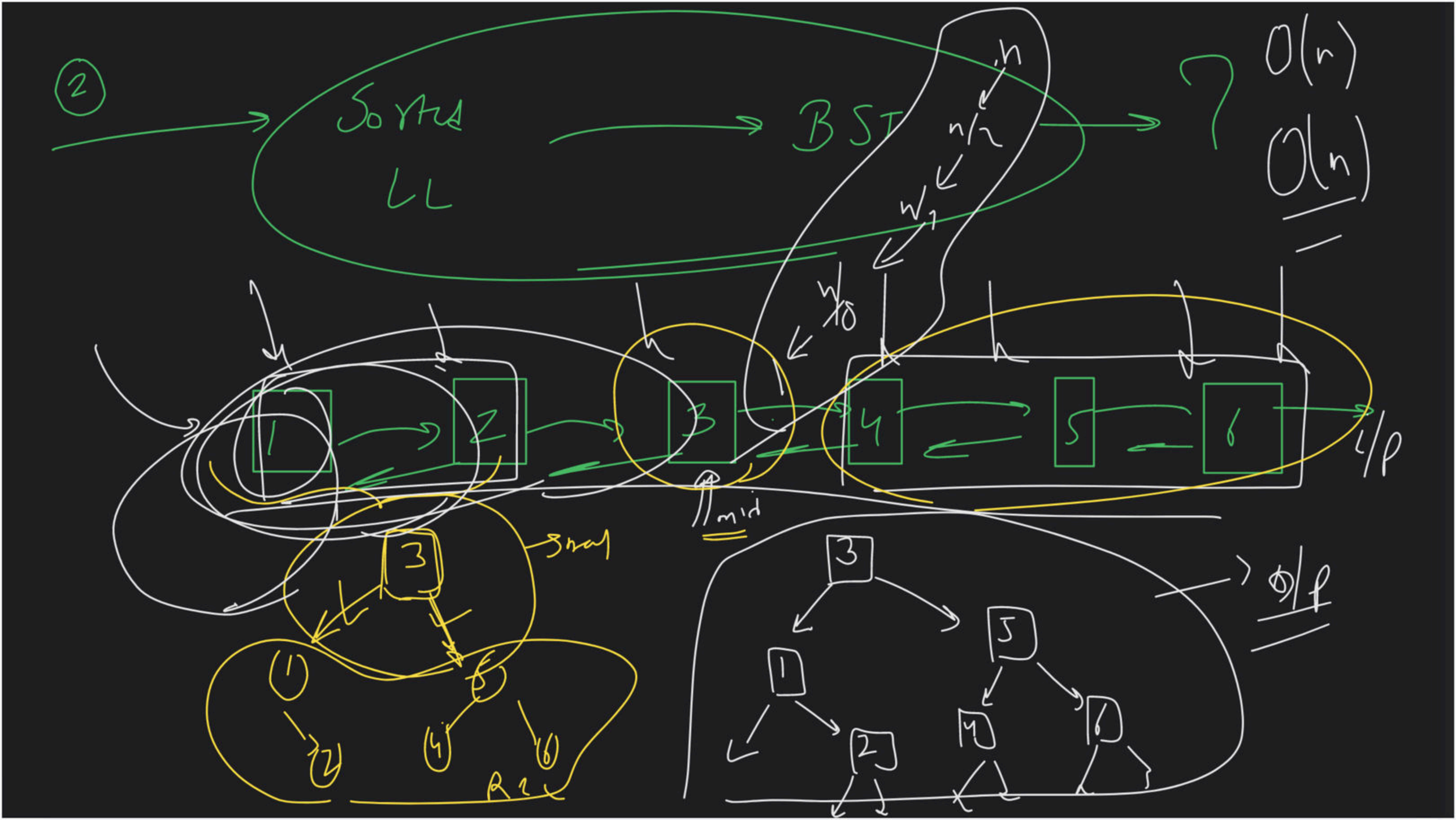
Love Babbar • May 20, 2023

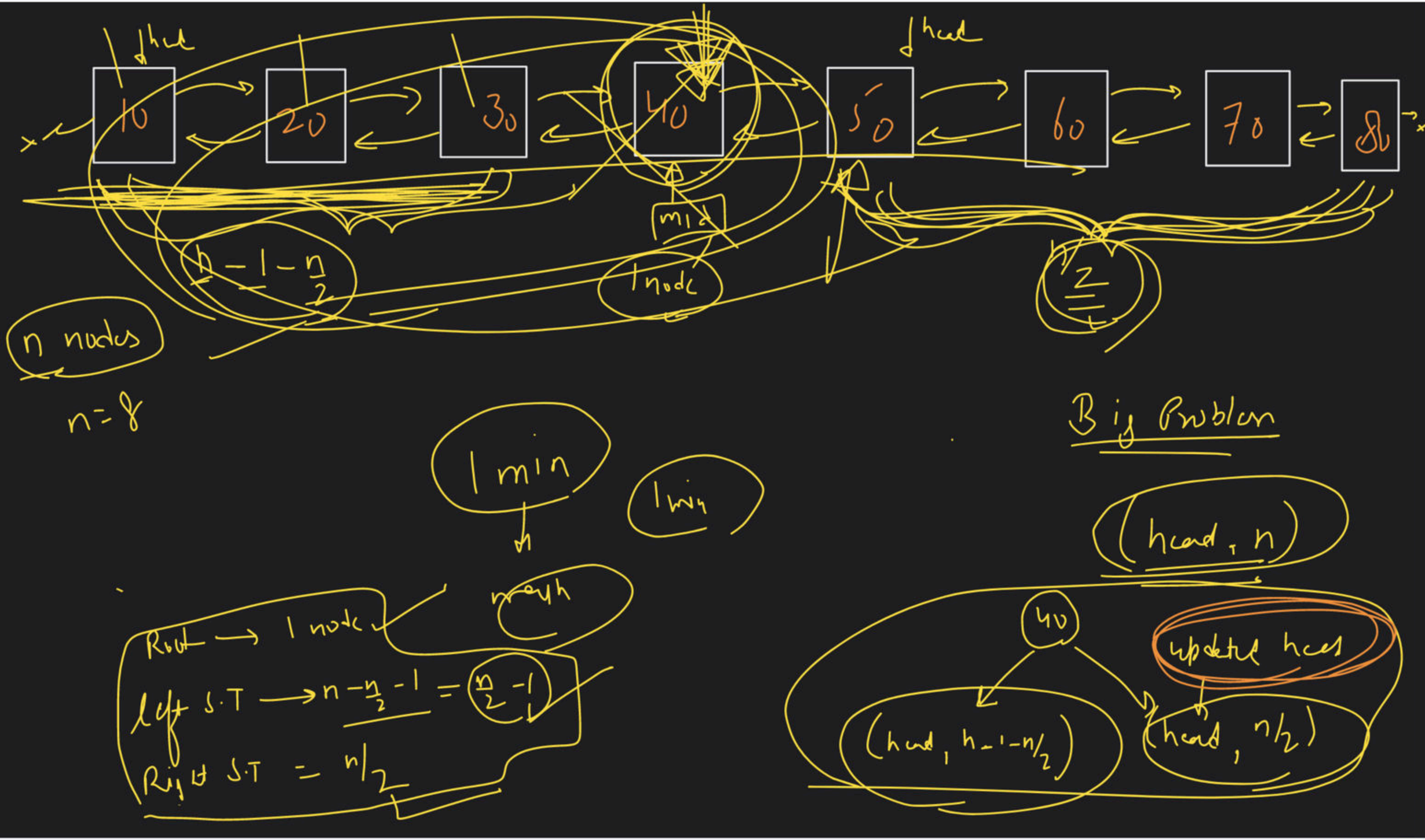

BST
convert
Sorted DLL
in-place
T.C → O(n)
S.C → O(H)
head
root
500
256
600
100
400
540
700
300
566
Rec
inorder
LNR
DLL
next
prev
left
RNL
Right subtree → Rec

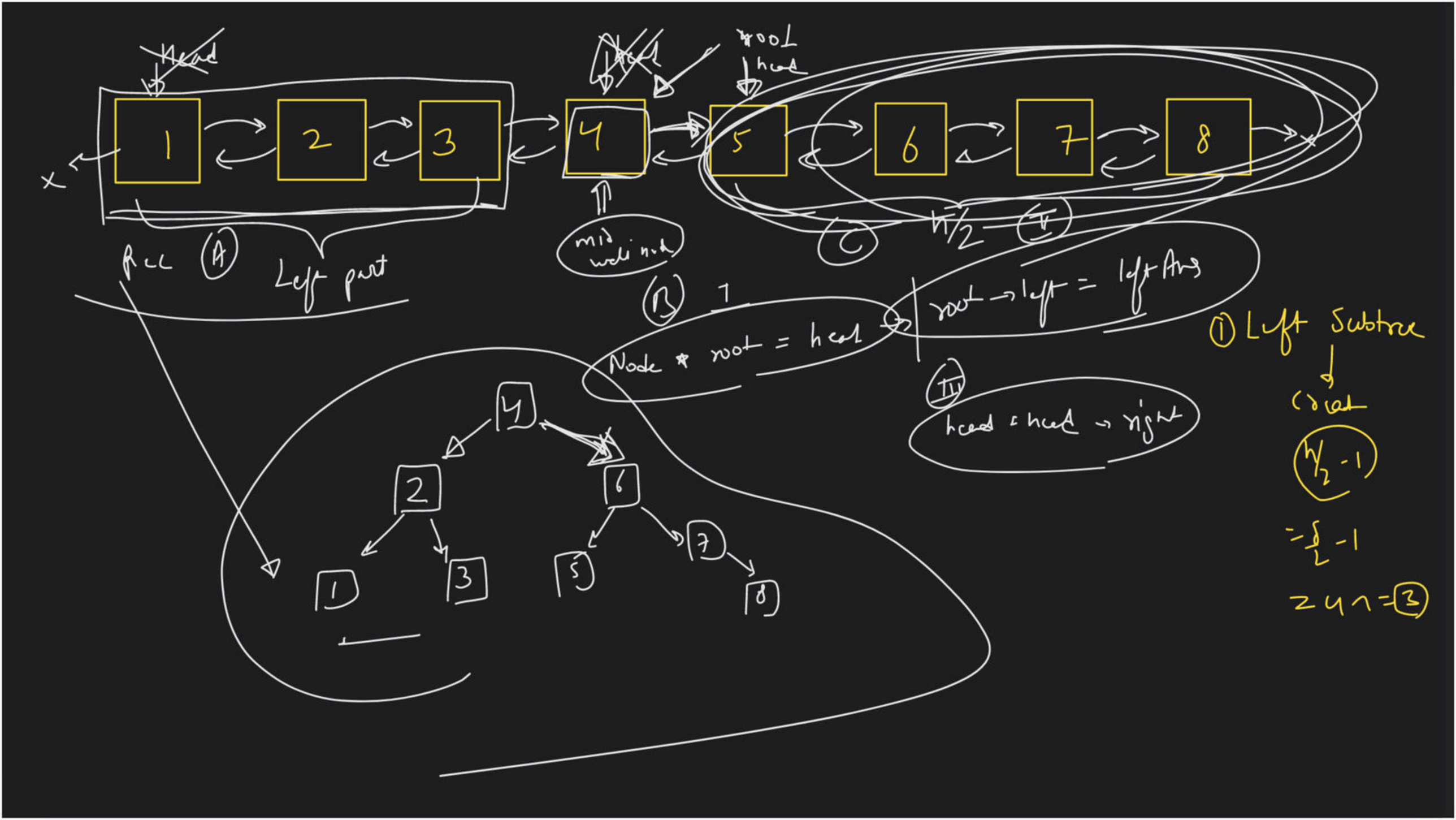

root
500
400
600
300
450
550
700
root
root
root
head
subtree
NUl
head
NUl
head
head
600
500
600
700

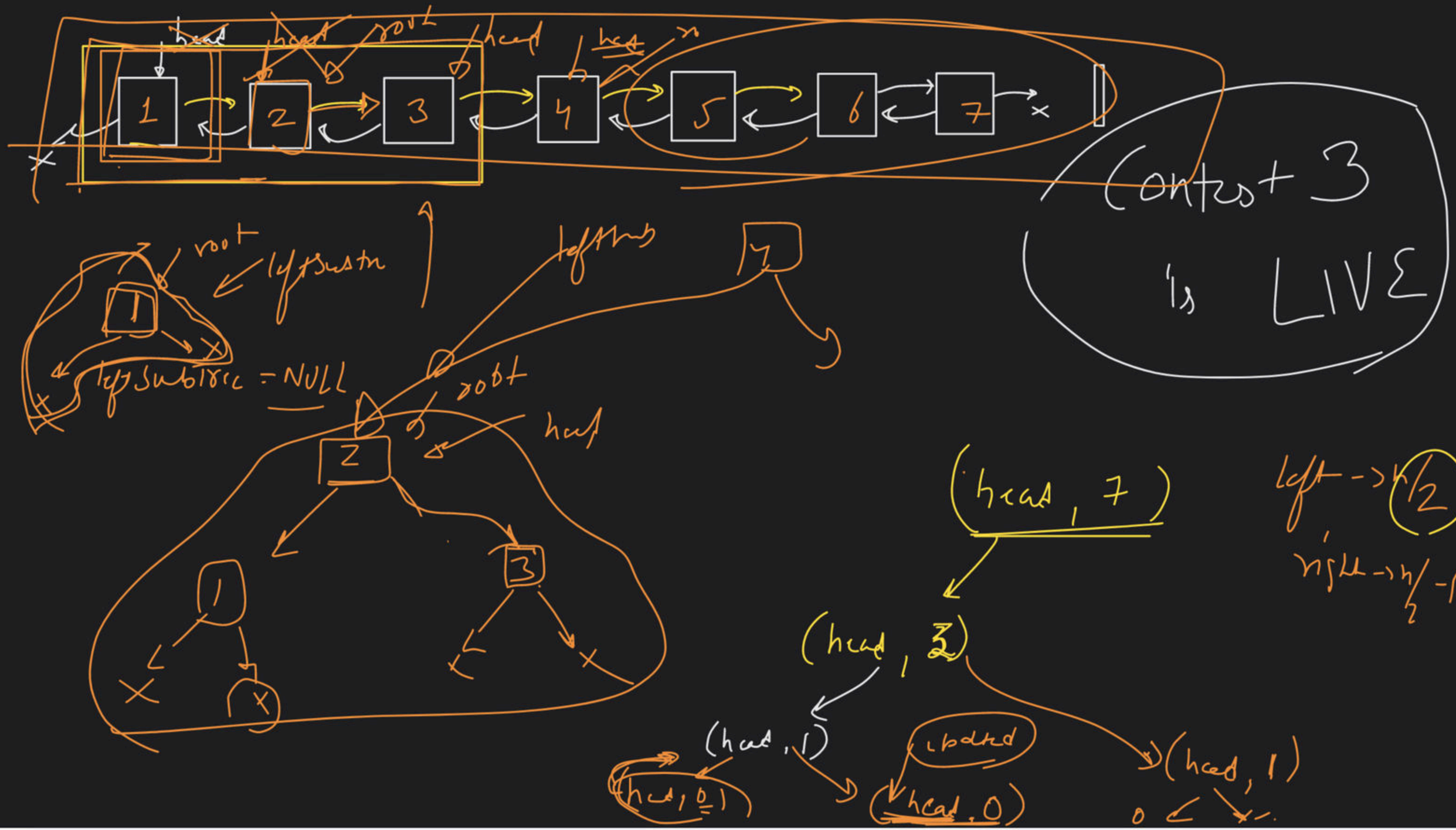

I/p -> BST -> Sorted DLL
root
L N R
head == NULL
head = root
head != NULL
RNL
NULL
head
5 7 8 10

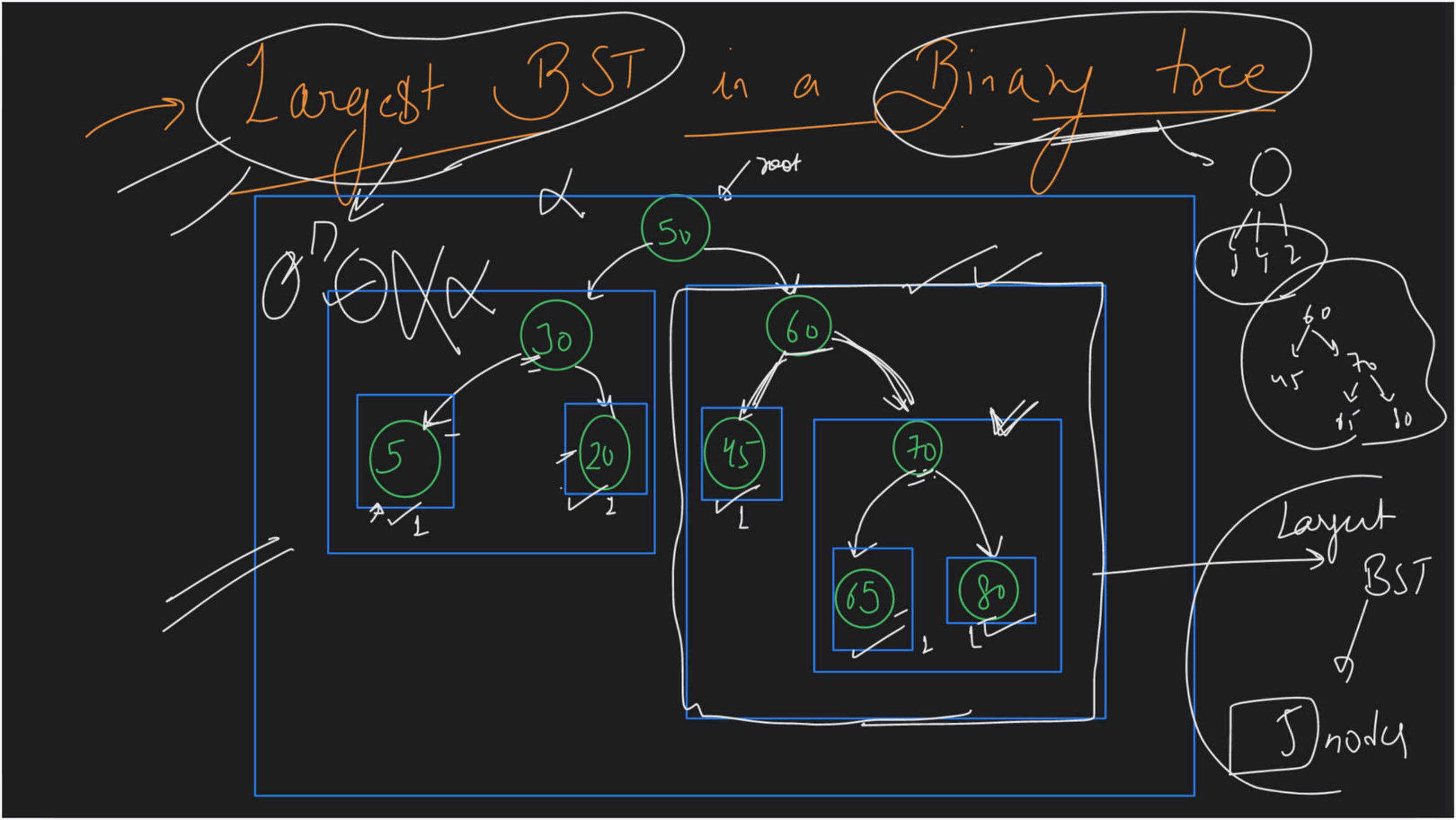

root
100
head
150 → 170 → 200 → 256
head
20 → 50 → 60 → 70
root
70
70
X
① Right Subtree
② not attach
root → right = head
head → left = root
③ Mid update
head = root
④ left subtree → LL
root → right = head
head → left = root
head = root
4-5

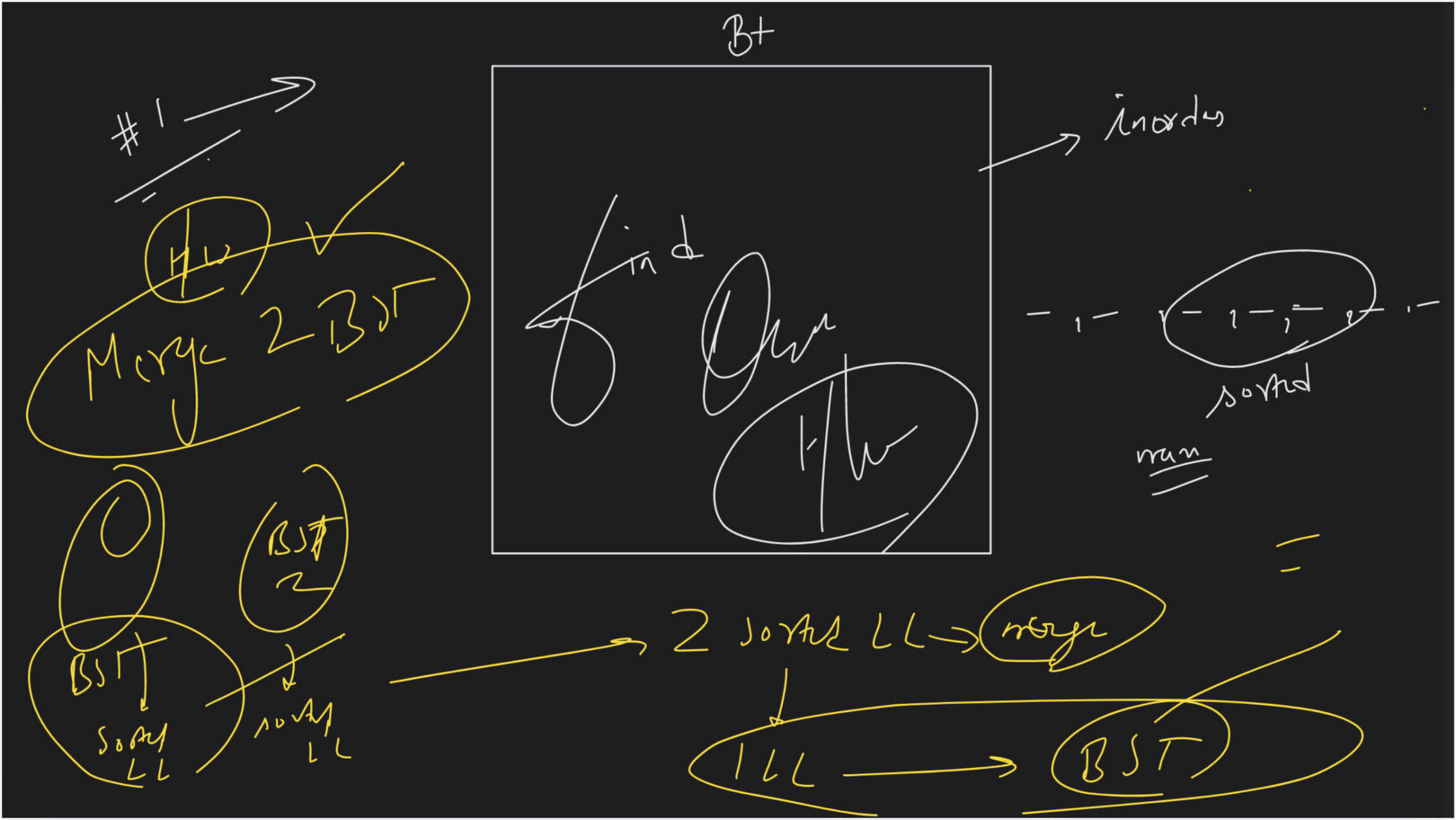

2
Sorted
LL
BST
n
n/2
n/4
?
O(n)
O(n)
1
2
3
4
5
6
mid
o/p
3
1
5
2
4
6

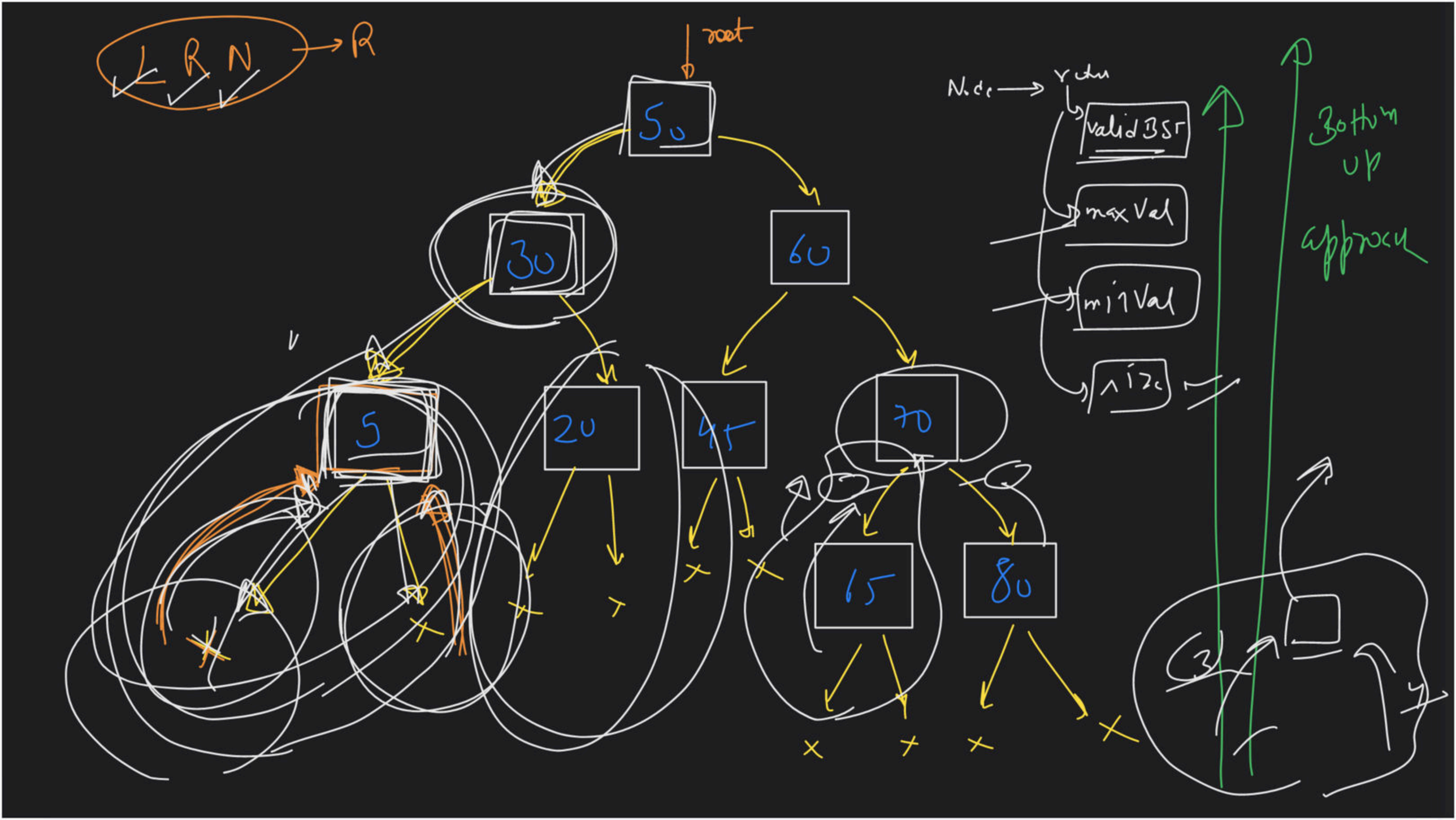

head
10
20
30
40
50
60
70
80
mid
n-1-n/2
1 node
n nodes
n=8
2
1 min
1 min
Root → 1 node
Left S.T → n-n/2-1 = n/2-1
Right S.T = n/2
Binary Problem
(head, n)
40
updated head
(head, n-1-n/2)
(head, n/2)

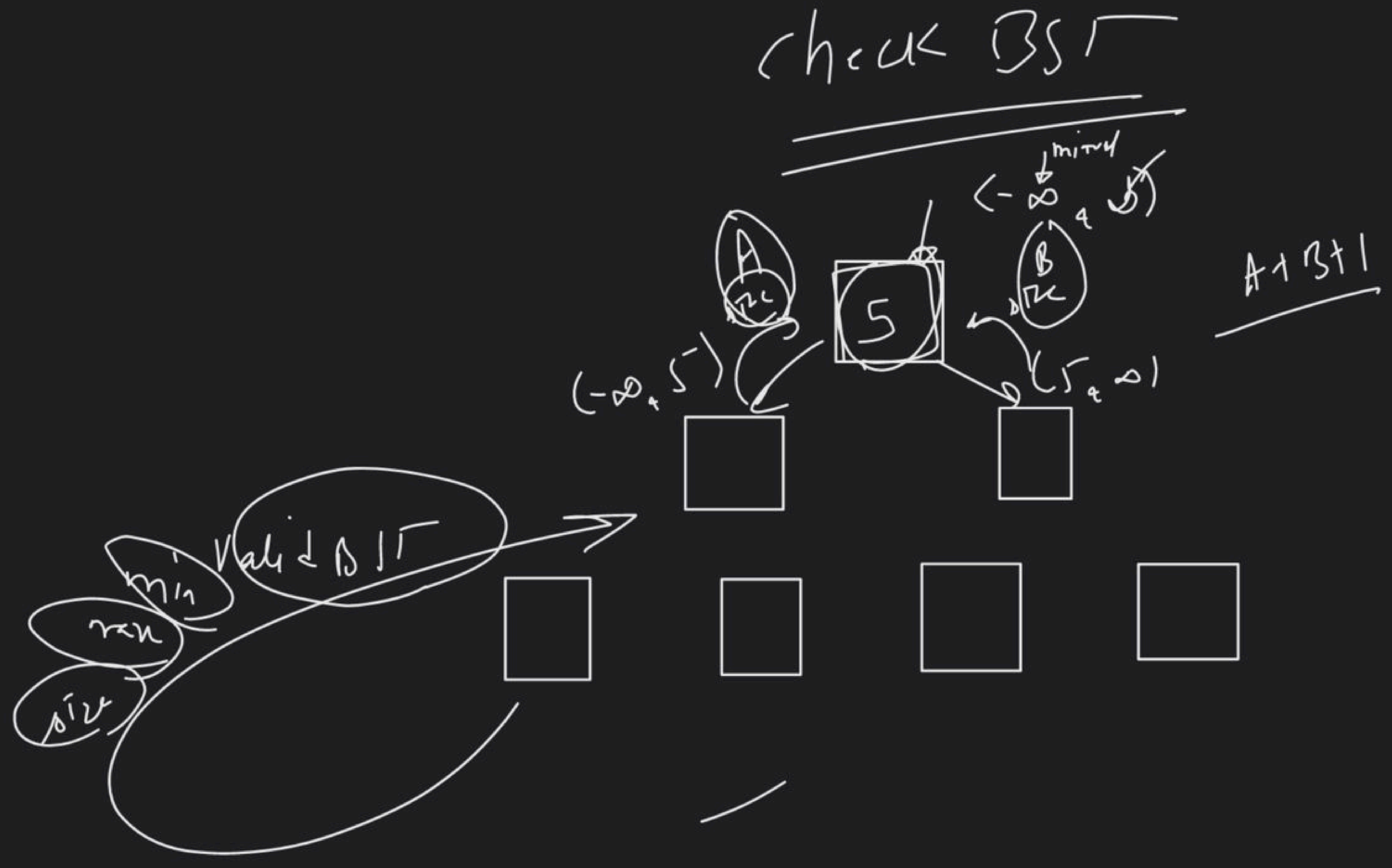

Head
root
head
1
2
3
4
5
6
7
8
x
mid
node
All (A)
Left part
(B)
Node * root = head
root → left = leftAns
(C)
n/2 = (I)
(II)
head = head → right
① Left Subtree
root
n/2 - 1
= 8/2 - 1
ans = (3)

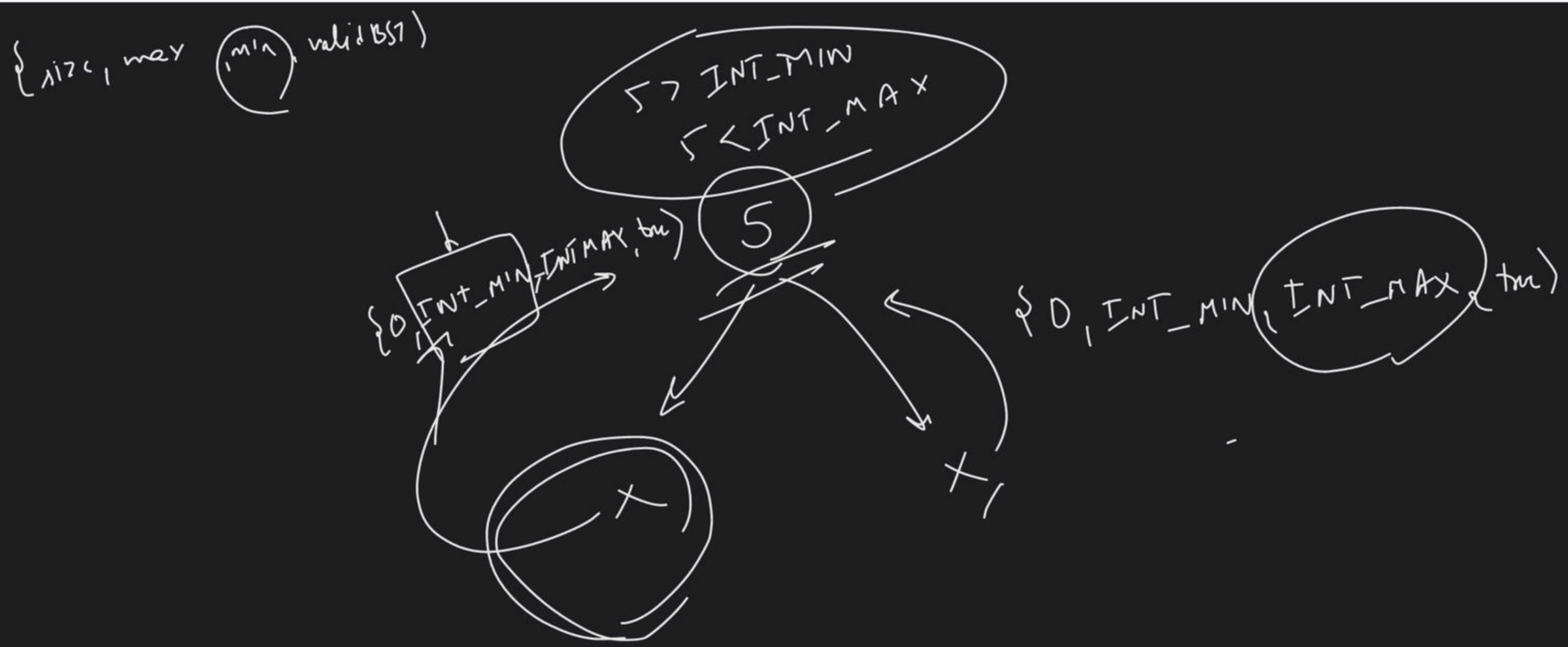

head
head
root
head
head
n
1
2
3
4
5
6
7
x
context 3
is LIVE
root
leftSubtree
leftSubtree
7
leftSubtree = NULL
root
head
2
1
3
x
x
(head, 7)
left -> n/2
right -> n/2 - 1
(head, 3)
(head, 1)
(head, 0=1)
(head, 0)
(head, 1)
0

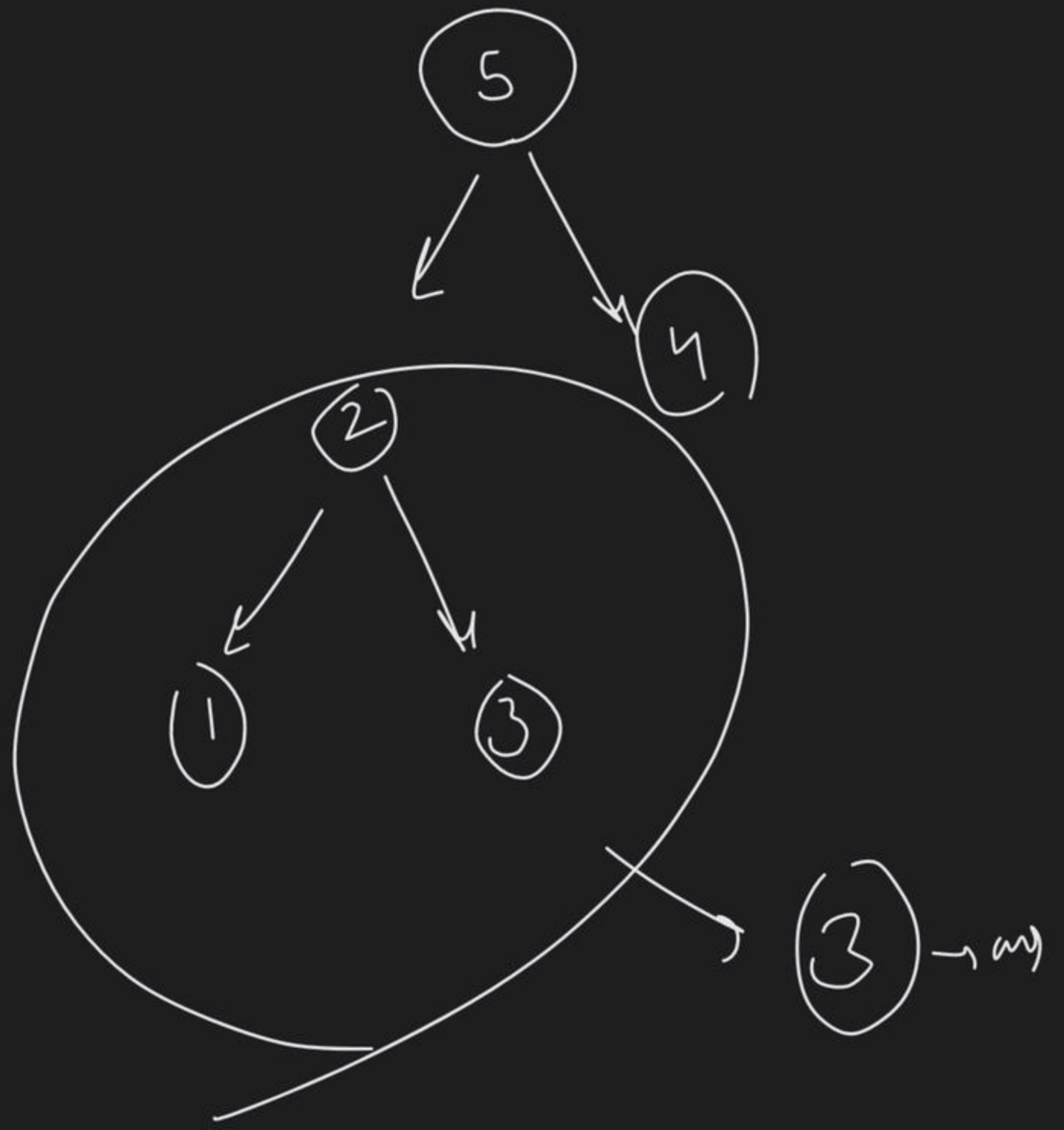

Largest BST in a Binary tree
root
50
30
60
5
20
45
70
65
80
Layout
BST
Node

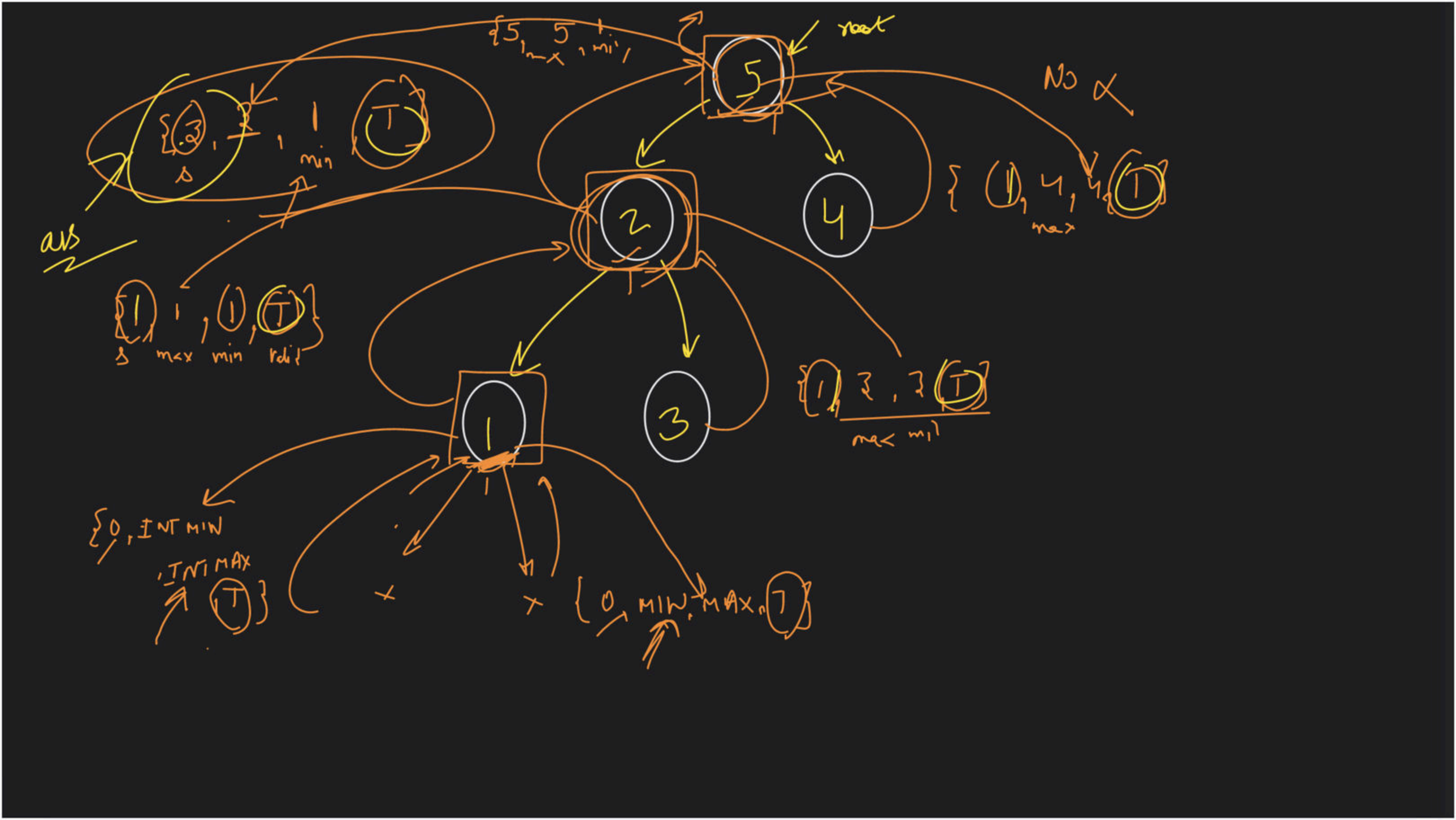

B+
#1
Merge 2 BST
find
inorder
sorted
run
BST
BST
sorted LL
sorted LL
2 sorted LL
merge
LL
BST

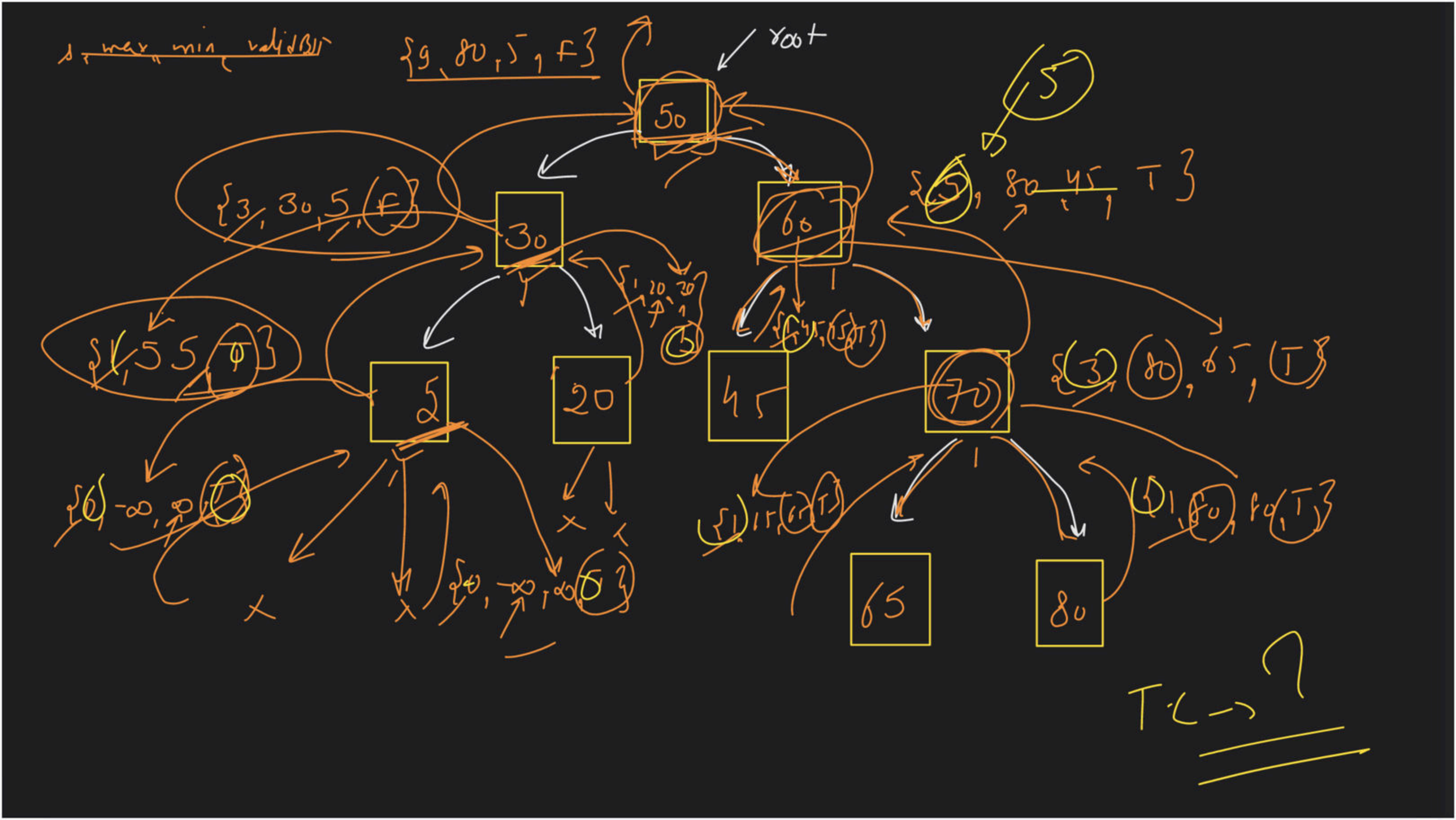

L R N
R
root
50
30
60
5
20
45
70
65
80
Node
valid BST
maxVal
minVal
size
Bottom up approach


check BST
A+B+1
5
valid BST
min
max
size


{size, max, min, validBST}
S > INT_MIN
S < INT_MAX
S
{0, INT_MIN, INT_MAX, true}
{0, INT_MIN, INT_MAX, true}
X
X


5
4
2
1
3
3


root
5
2
4
1
3
min
max
ans
x
x


root
50
30
60
5
2
20
45
70
65
80