BST → Binary Search Tree value based data structure.

BST → left subtree max < node.data < right subtree min

Node → (1) data (2) left (3) Right

* follow by Every node.

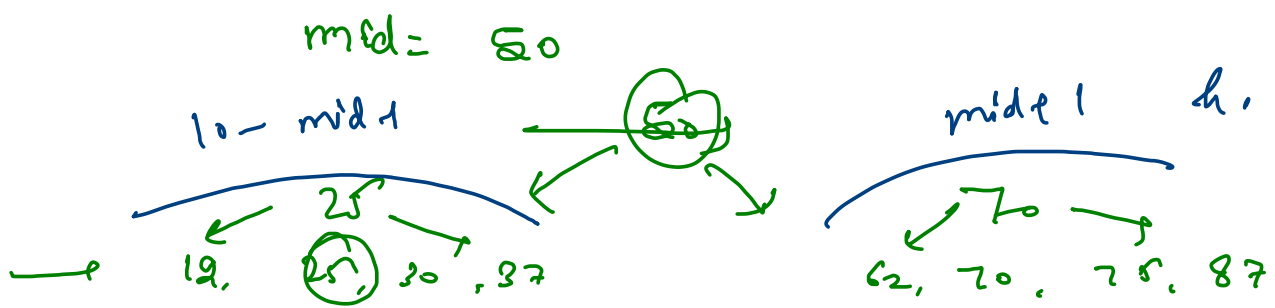
* In Order in the BST is sorted.

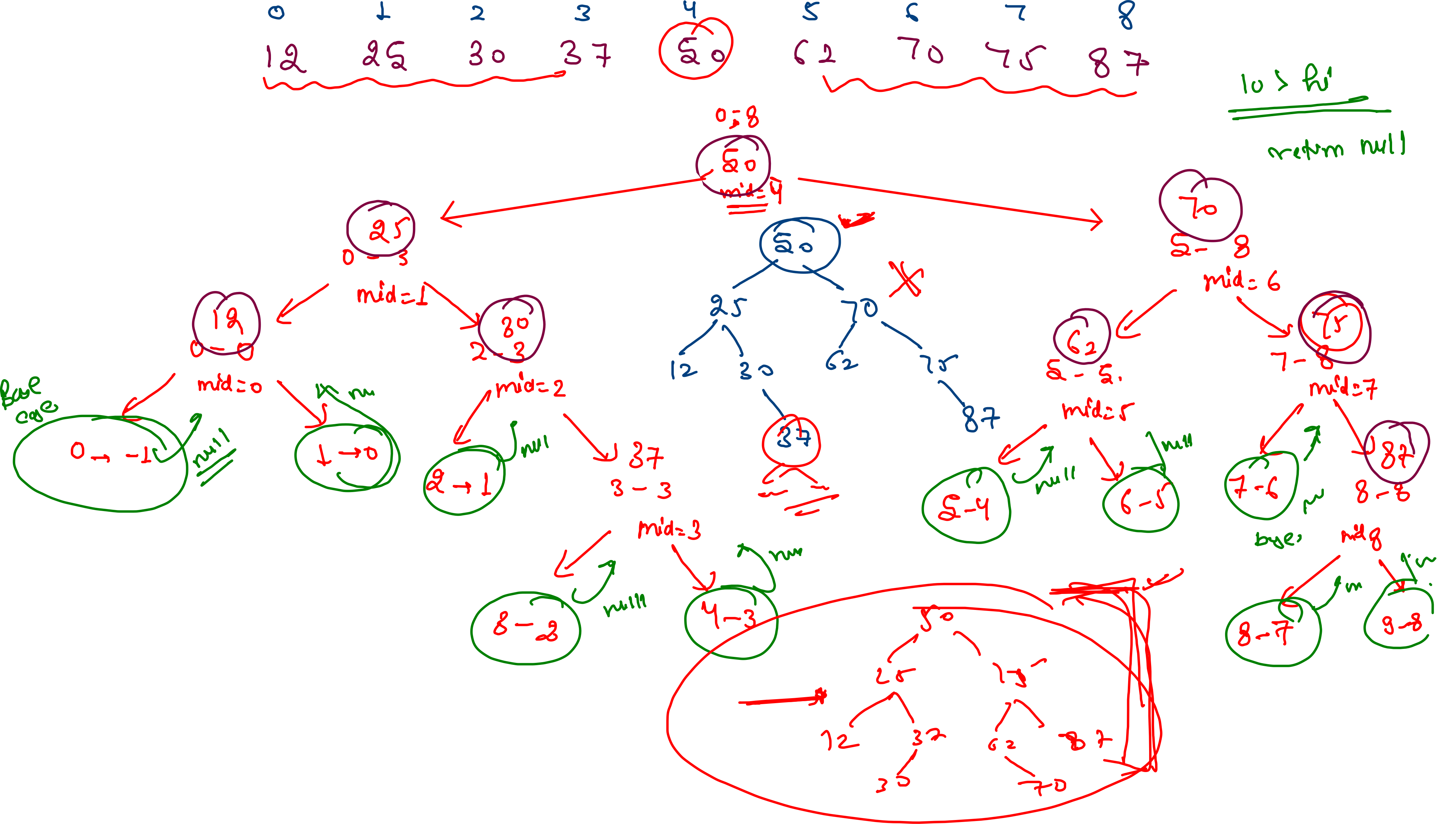
Construction.

lo ↓ hi ↓

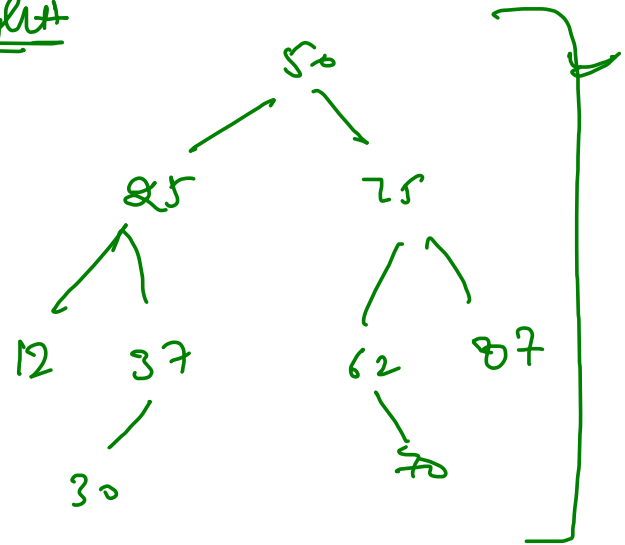
InOrder → 12 25 30 37 50 62 70 75 87

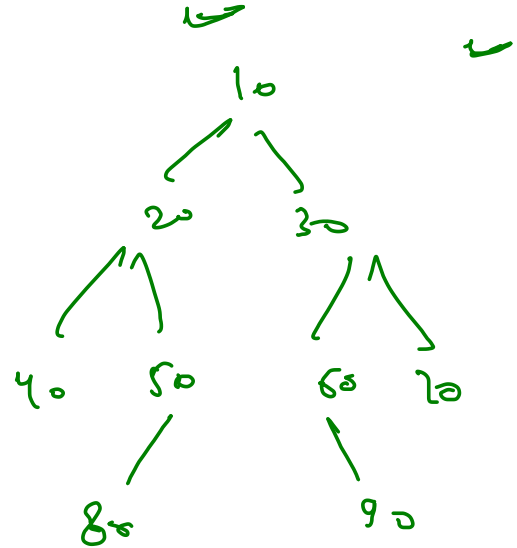
Binary search




0 1 2 3 4 5 6 7 8
12 25 30 37 50 62 70 75 87
lo > hi
return null
0-8
50
mid=4
25
0-3
mid=1
12
0-0
mid=0
Base case
0 → -1
null
1 → 0
30
2-3
mid=2
2 → 1
null
37
3 - 3
mid=3
3 - 2
null
4 - 3
70
5- 8
mid=6
62
5 - 5
mid=5
5-4
null
6-5
null
75
7-8
mid=7
7-6
87
8-8
mid 8
8-7
9-8
50
25
70
12
30
62
75
37
87

height





same for BST

height → Structural problem

size → " "

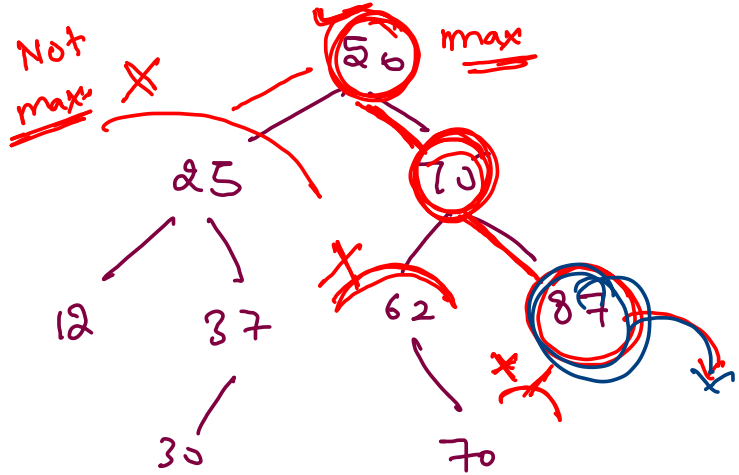
Diameter → " "

Max → Value based problem

min → "

sum → Not sum

why not login ??



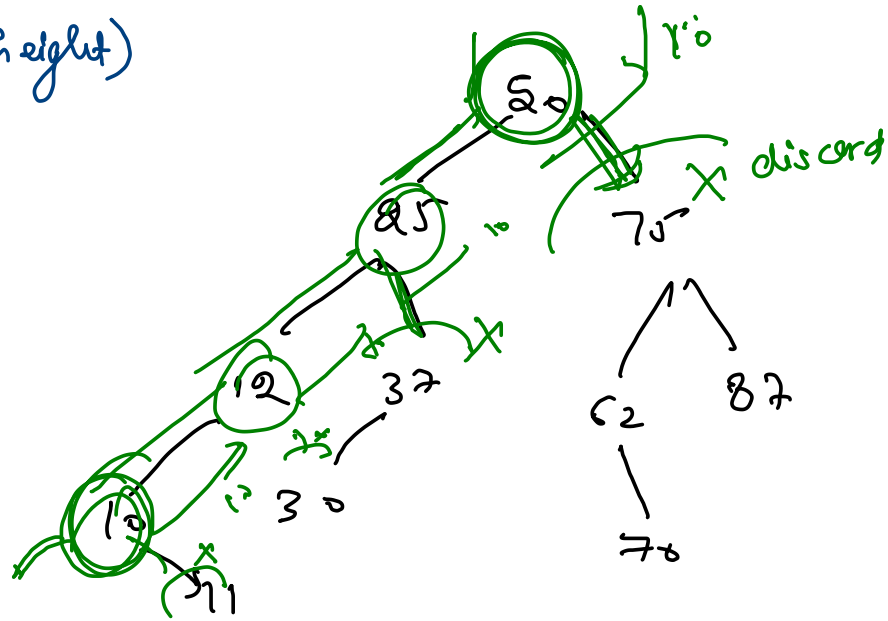
max =?

max

node.right = = null

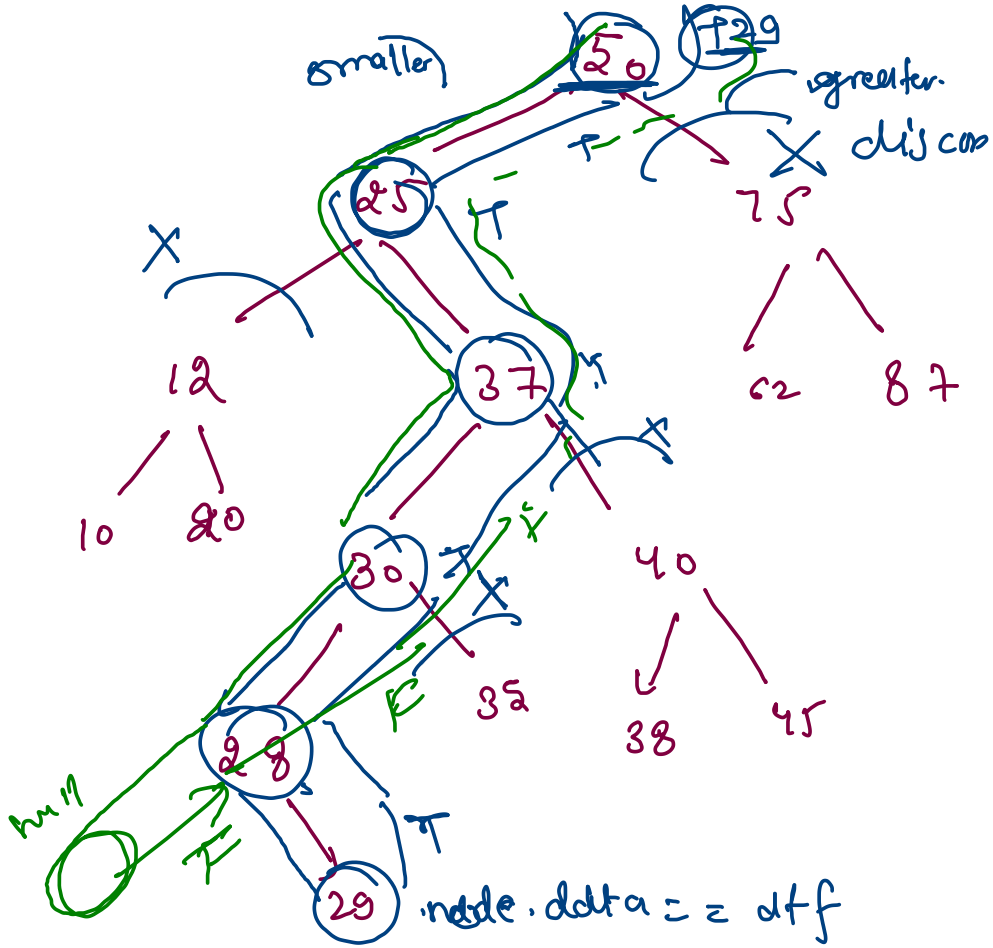
return node.data;

complexity → O(height)



min

complexity - O(height)



dtf = 29

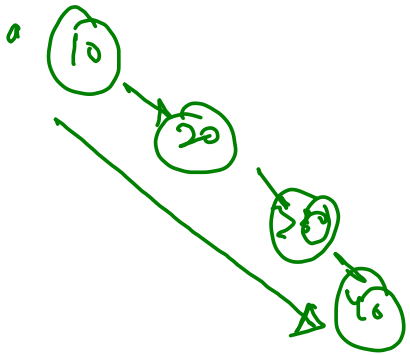
(data to find ≡ dtf)

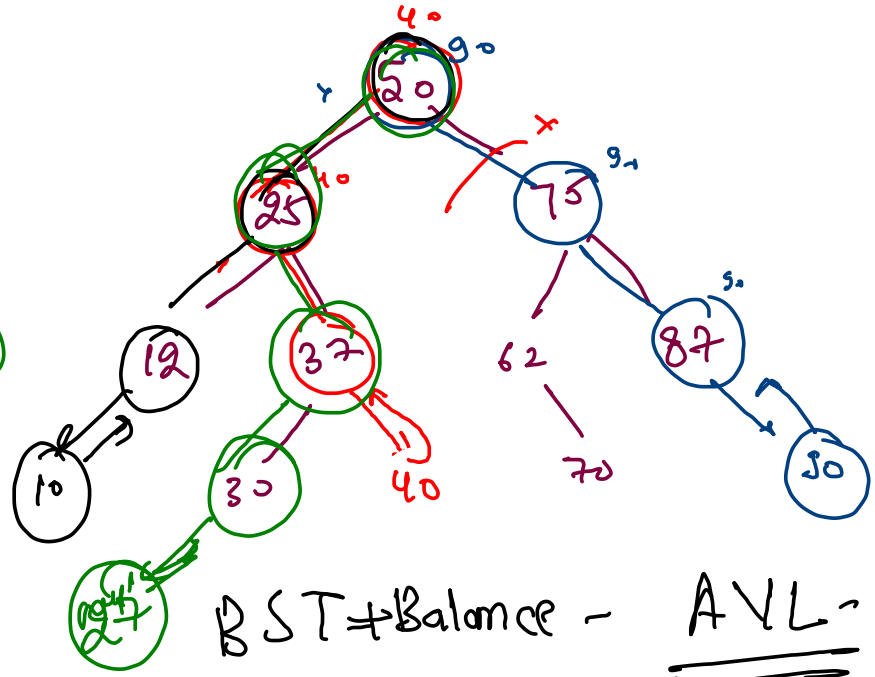
dtf = 27

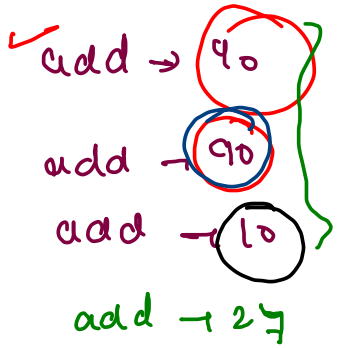
if null

is encounter

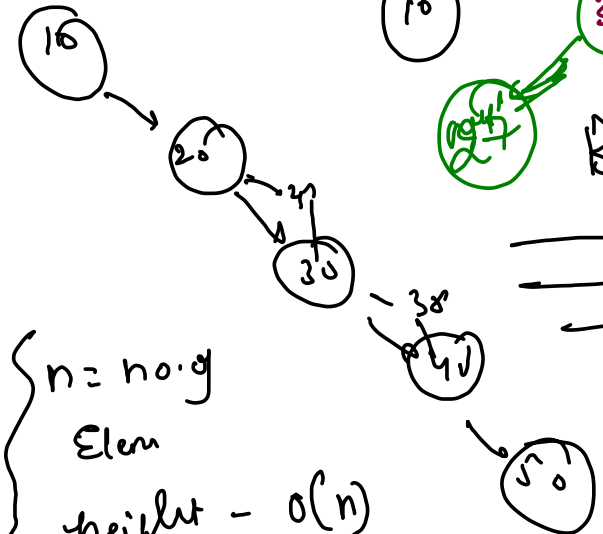
that means data is not present







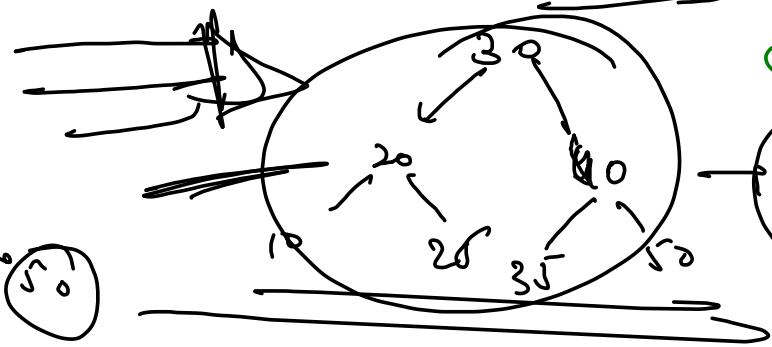
addition → when we reach at null #



n = no. of Elem

height - O(n)

BST + Balance - AVL -



Complexity → O(height)