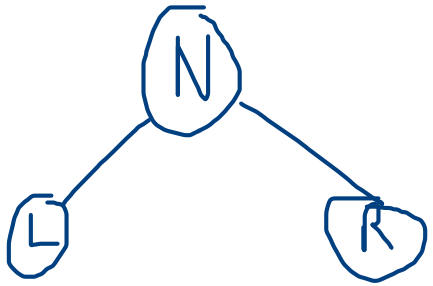
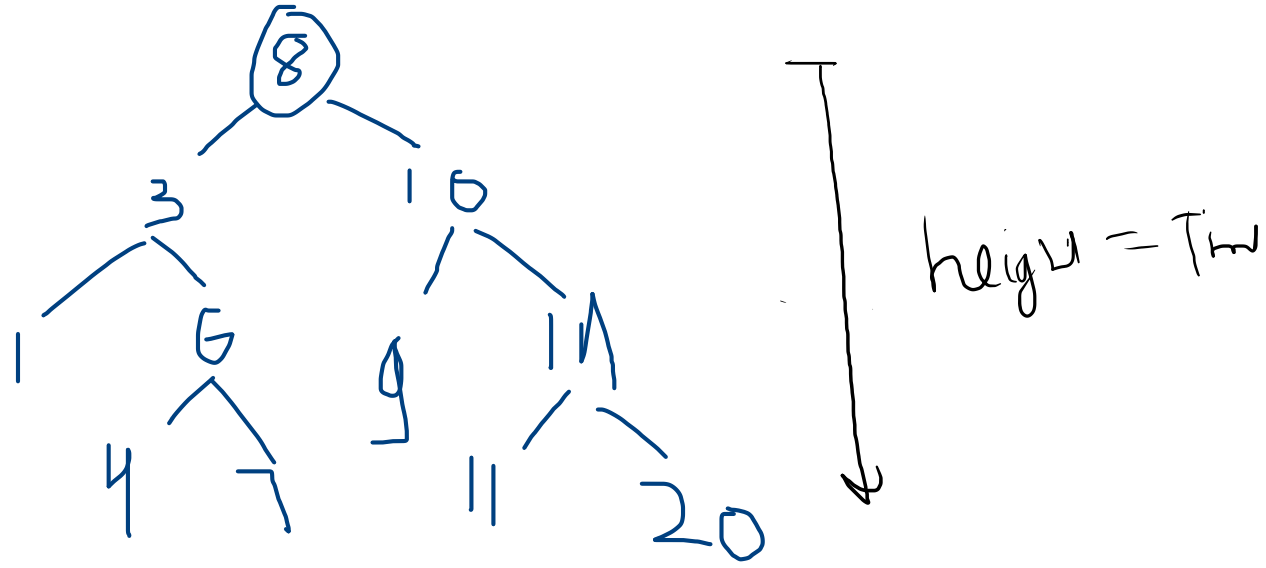
Binary Search Tree



$L < N < R$

BST



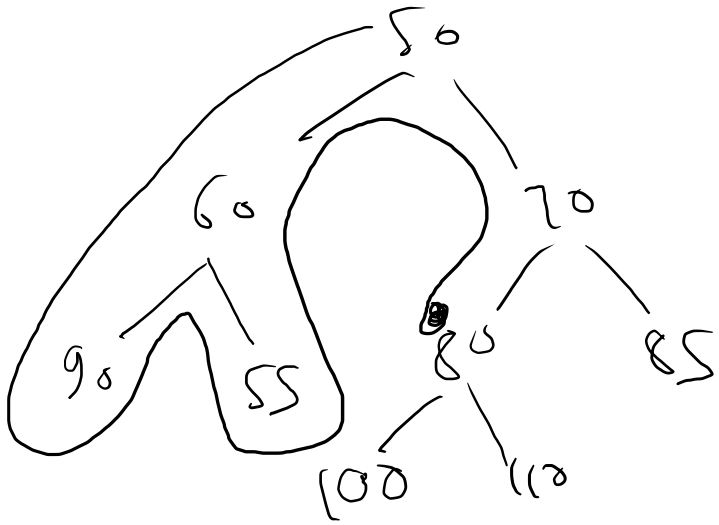
more → Right

min → Left

Searching → faster

Search (9) → O(h)

BT =>



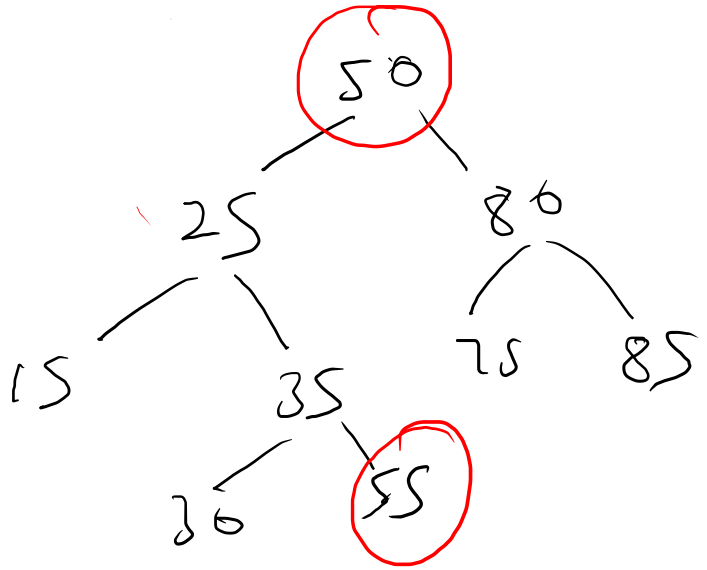
n => no. of nodes

Search(80) => O(n)

BST

BT

Eg



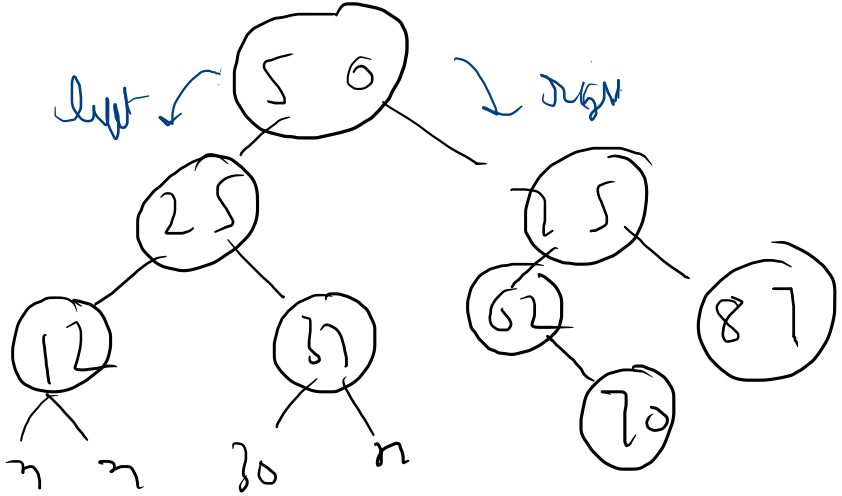
55 >= 50
=> BT

all of left tree nodes < Root Node / Parent

all of Right tree nodes > Root Node / Parent

Size, Sum, Max, Min

BST



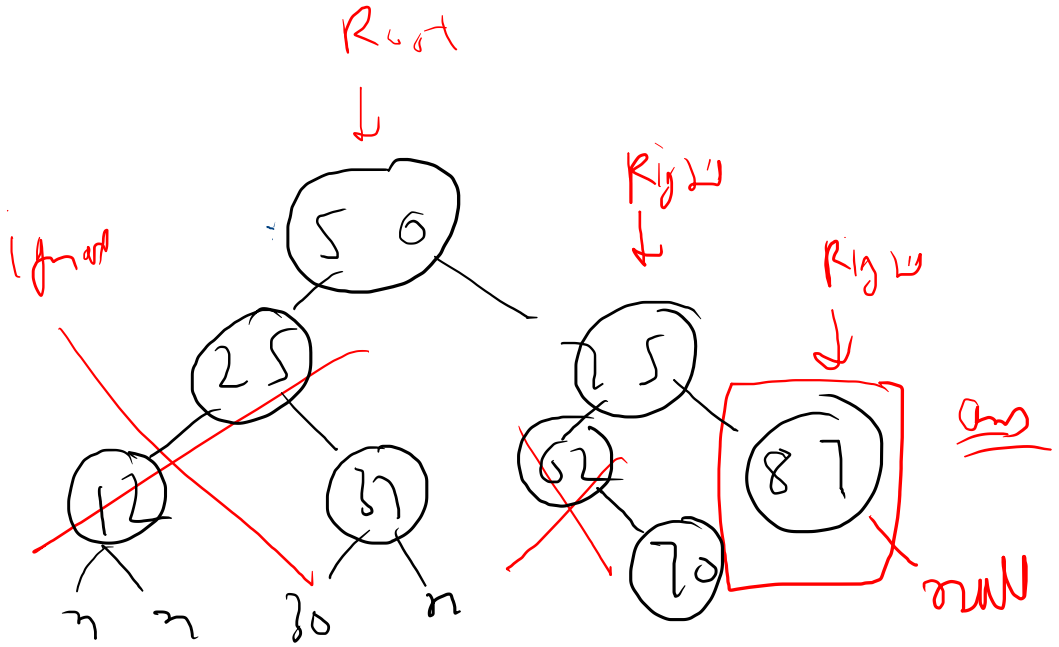
Size => no. of nodes

=>

left + right + 1

Sum

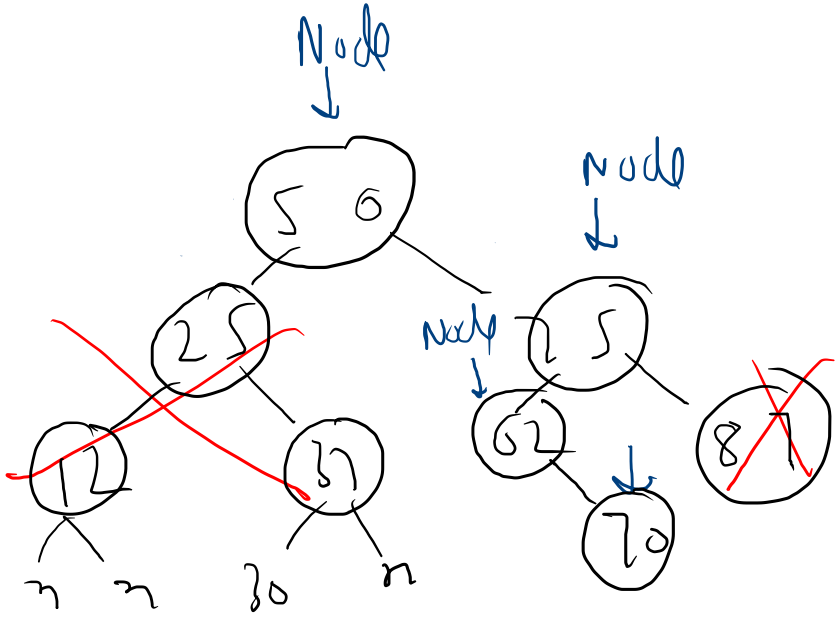
left + right + nodeVal



root
parent
node
50
25
75
12
37
62
87
70

Find

Search node => 70



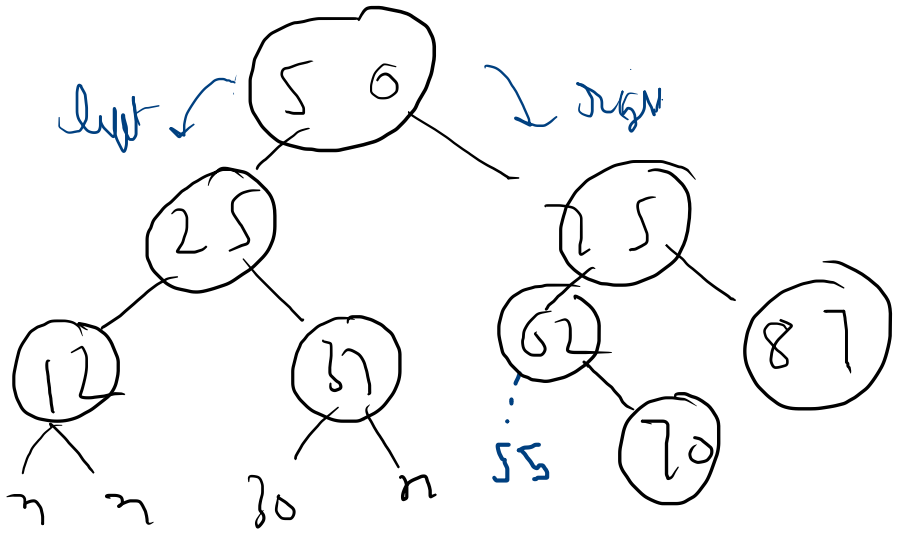
(1) 70 > 50 → go Right

(2) 70 < 75

(3) 70 > 62

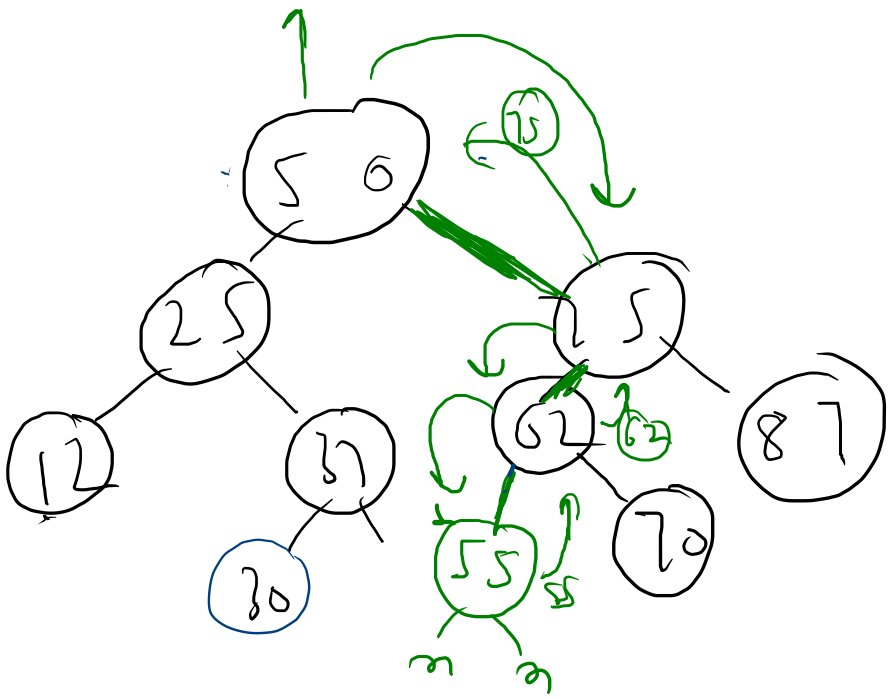
(4) 70 == 70 → true

ADD in BST



55

node to return



55

50 < 55

→ right

55 < 75

62 > 55

left == null ⇒ create node ⇒ 55