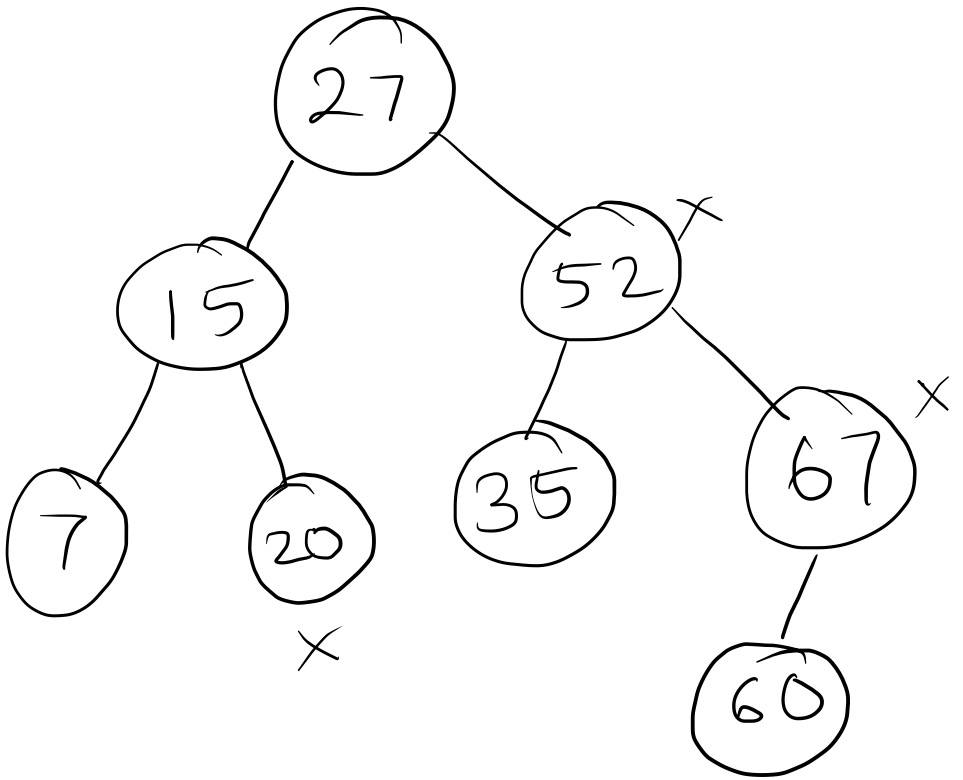
BST

1. checkBST
2. Search in BST
3. Insert in BST
4. Delete in BST

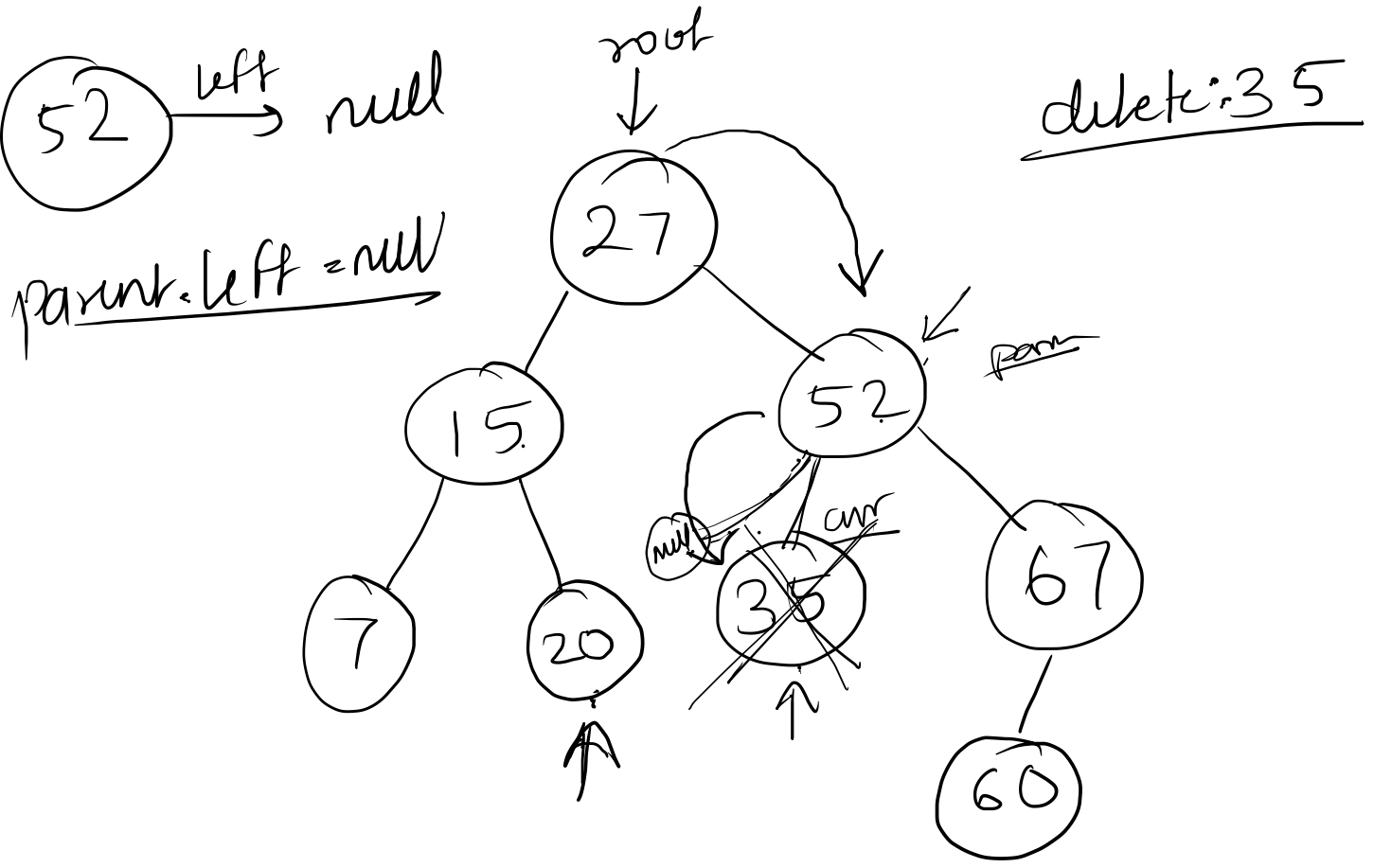
Deletion in BST



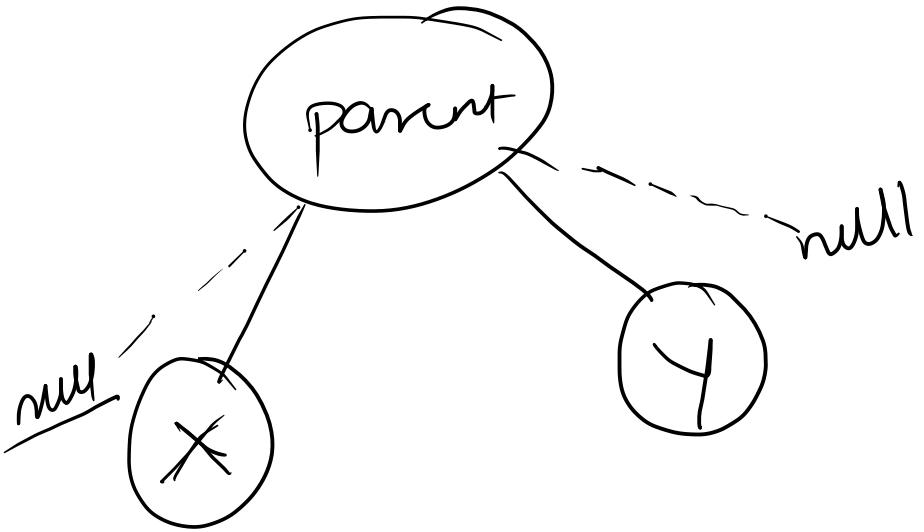
Deletion

1. Search the node
2. delete the node

Deletion in BST



Track parent node



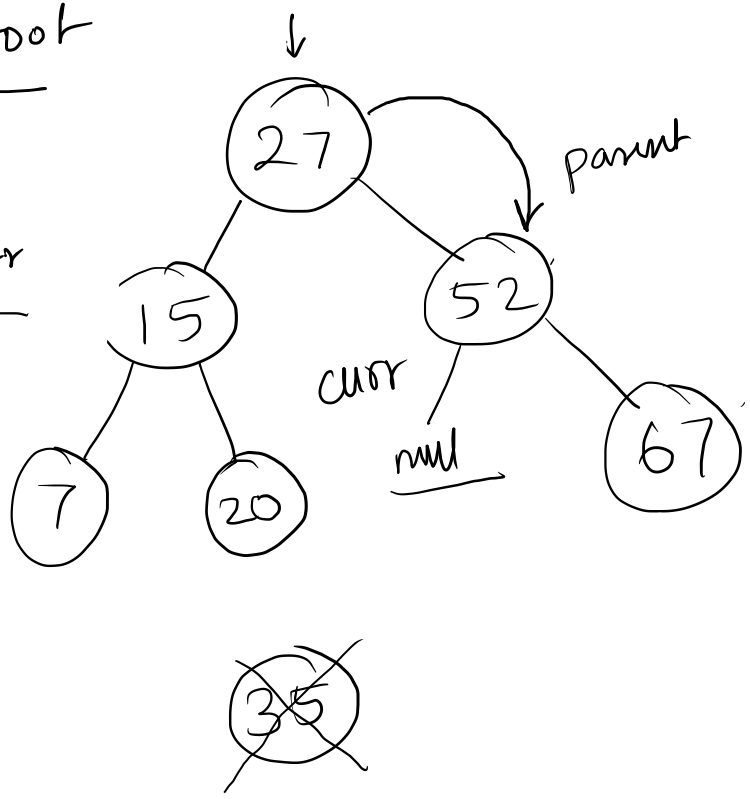
delete X → parent.left = null

delete Y → parent.right = null

parent = null

curr = root

delete : 35



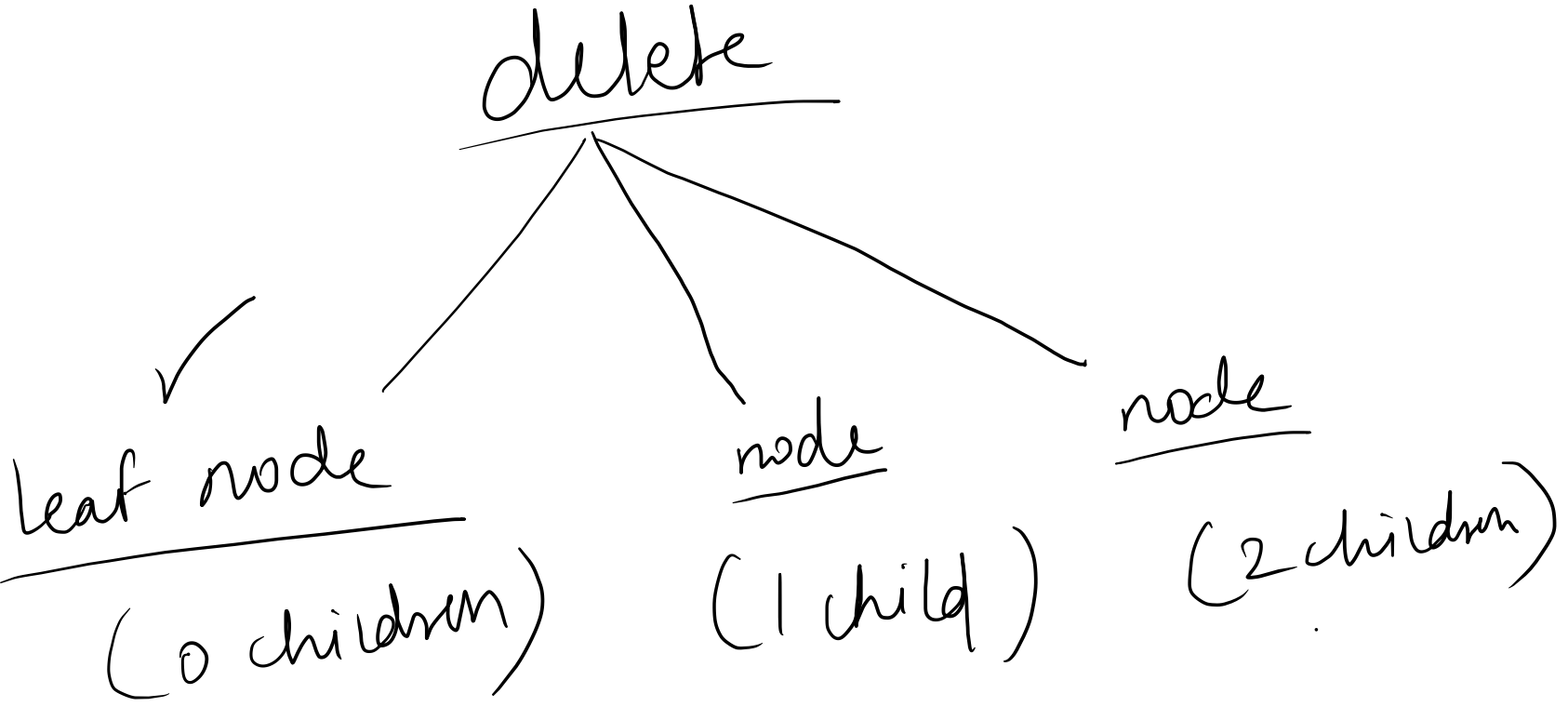
parent = curr

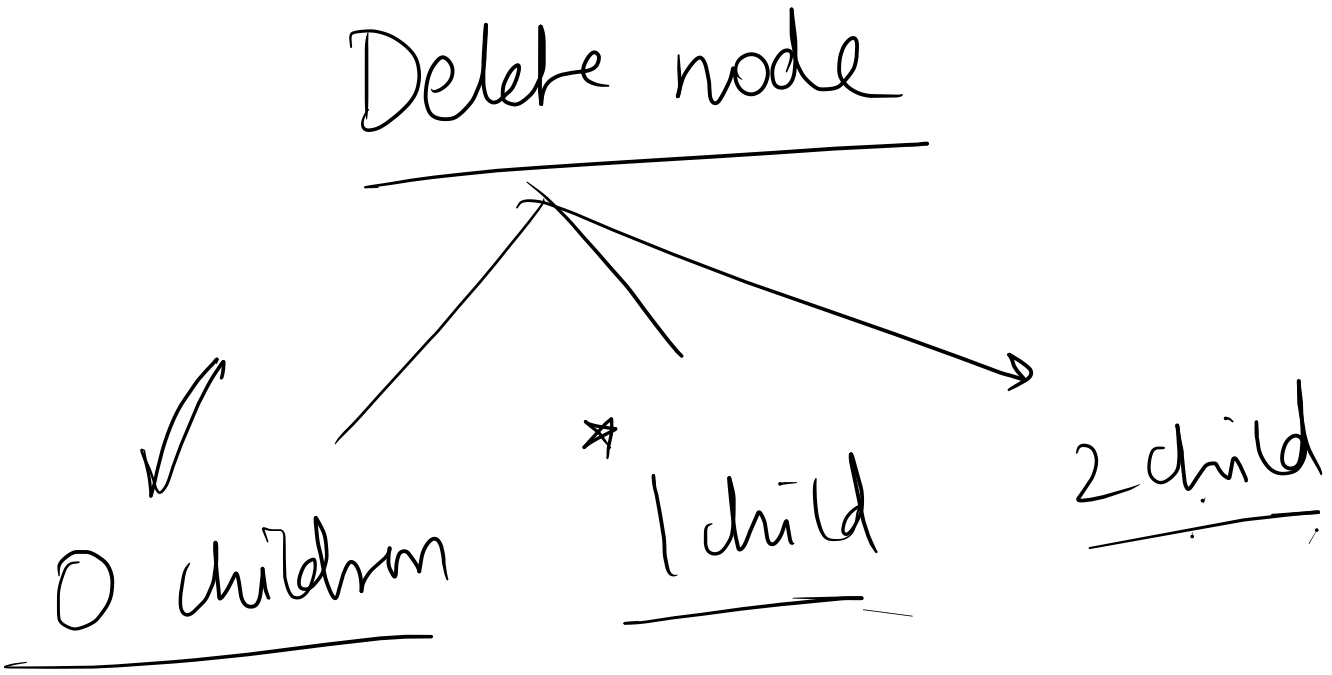
if (parent.left = curr)

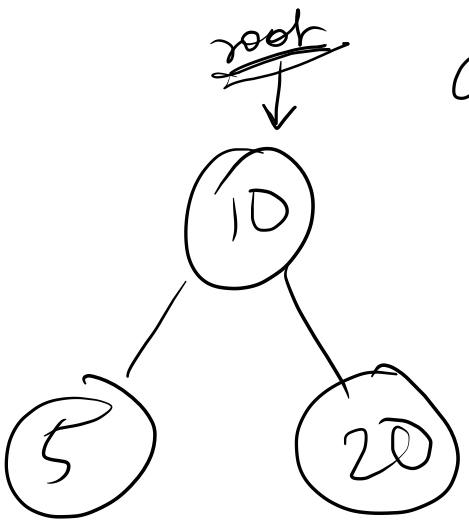
parent.left = null

else

parent.right = null

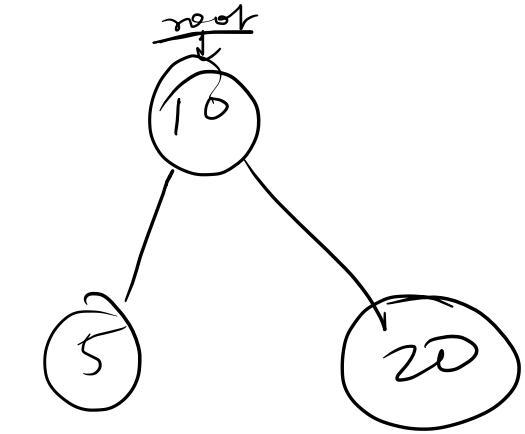

delete
leaf node
(0 children)
node
(1 child)
node
(2 children)

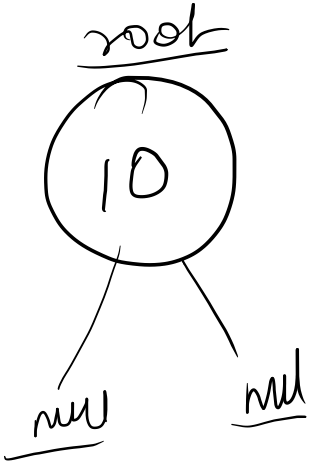

Delete node
0 children
1 child
2 child

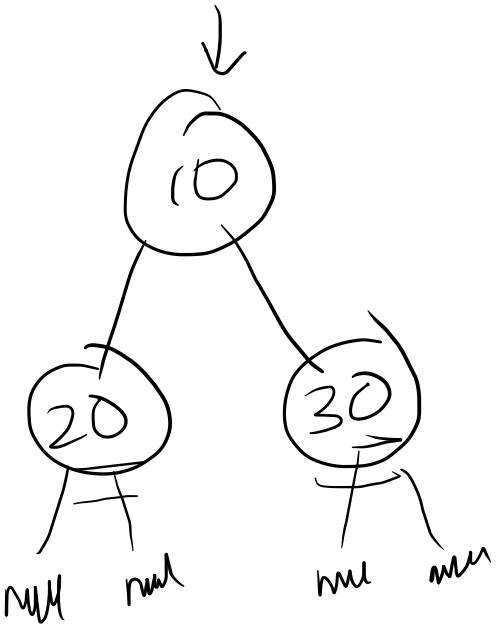


delete 3

delete 3







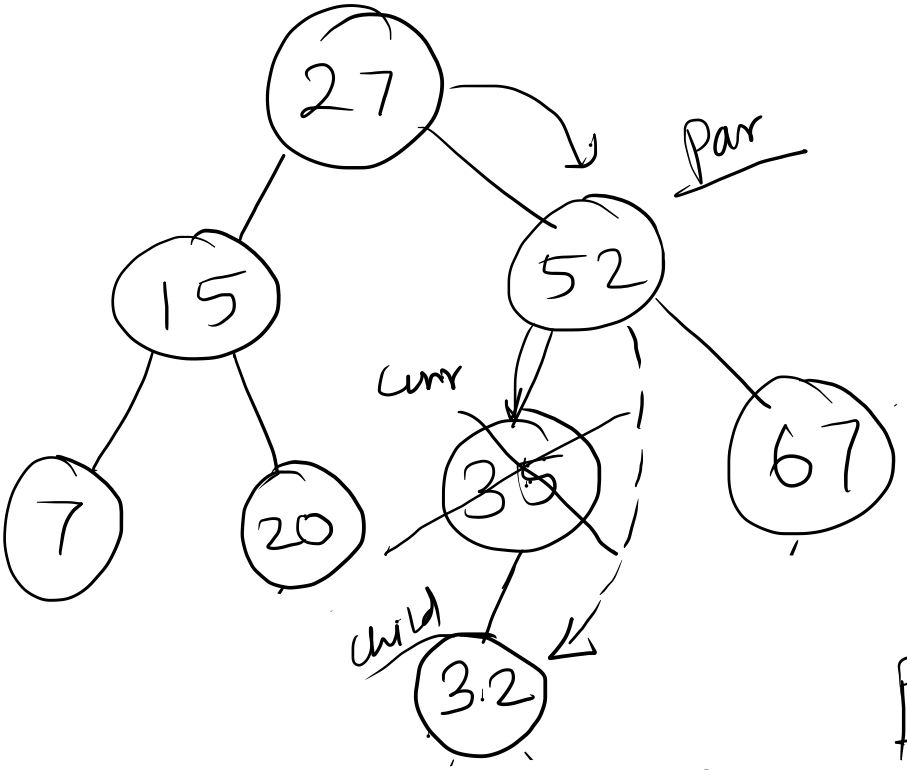
root → leaf

root
10

root
null

delete node with 1 child

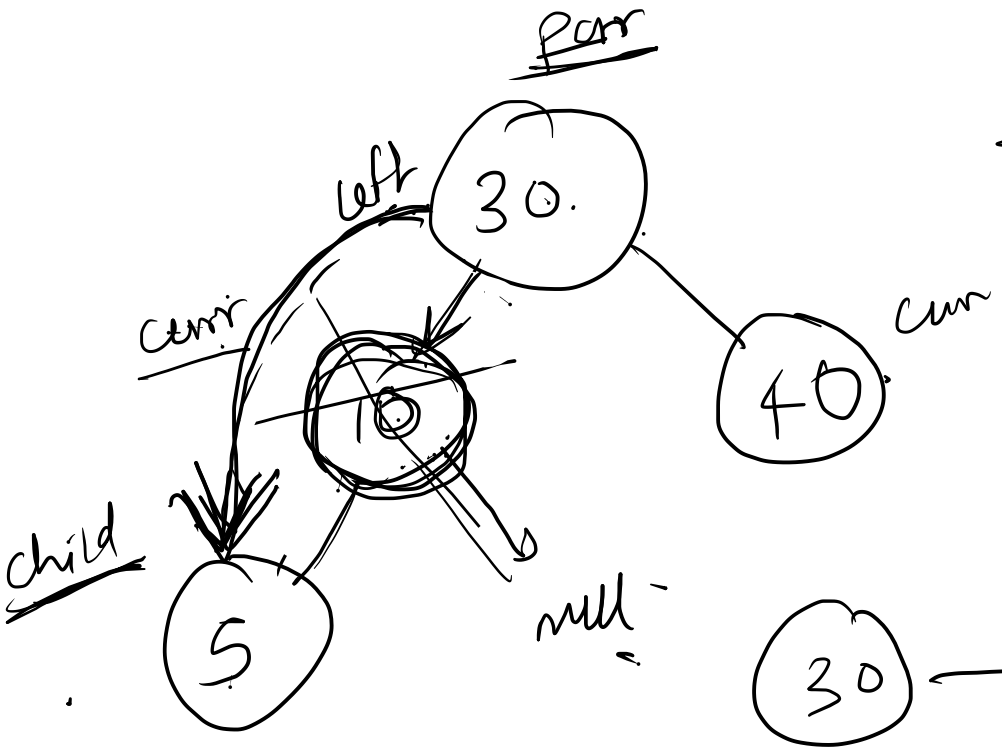
Node with 1 child



delete 35

parent.left = child

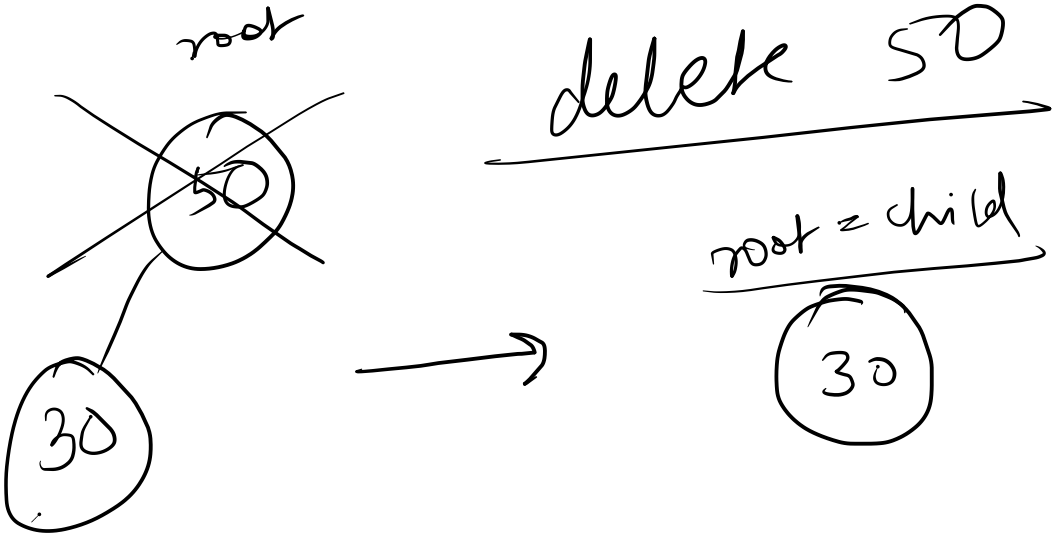
parent.right = child

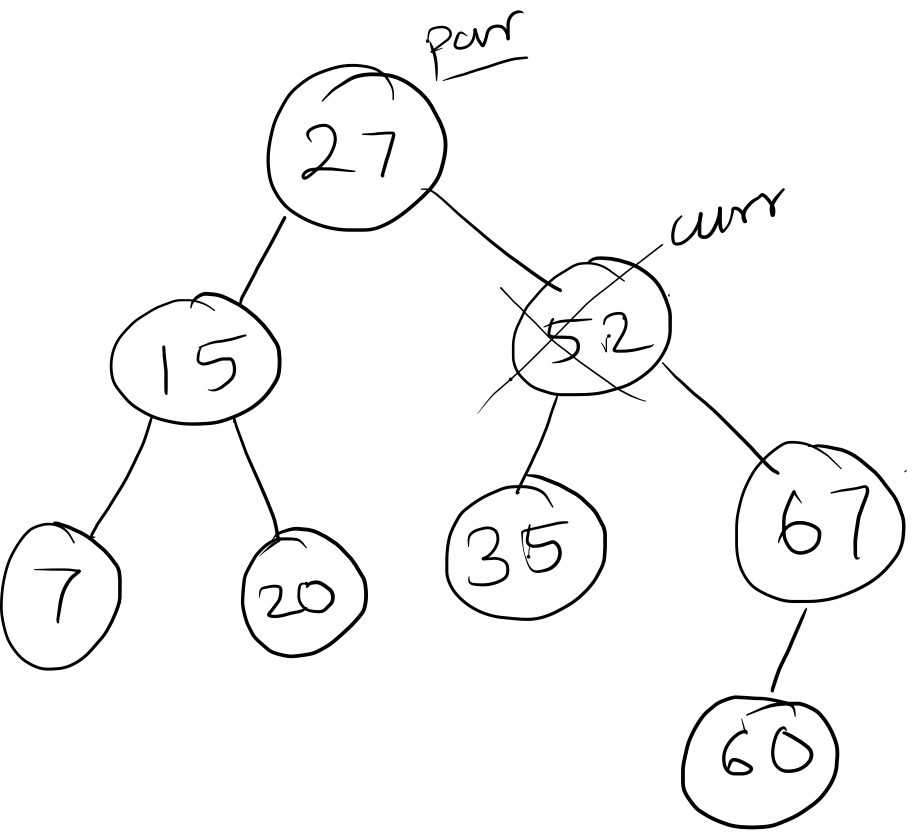


delete 10

30 → left → 5

par → left → child

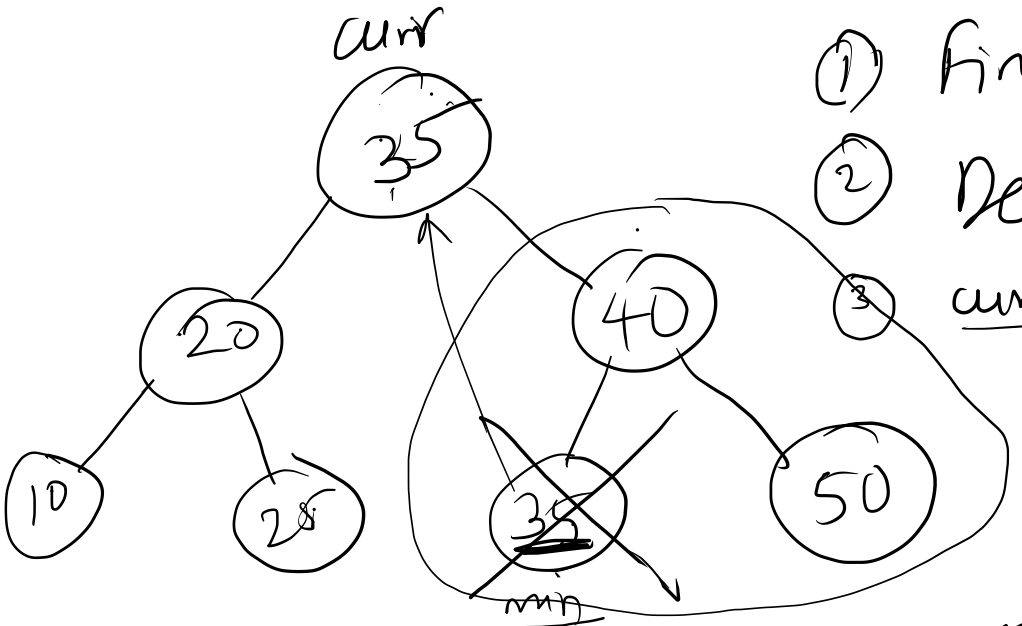

root
50
30
delete 50
root = child
30



delete: 52

left max

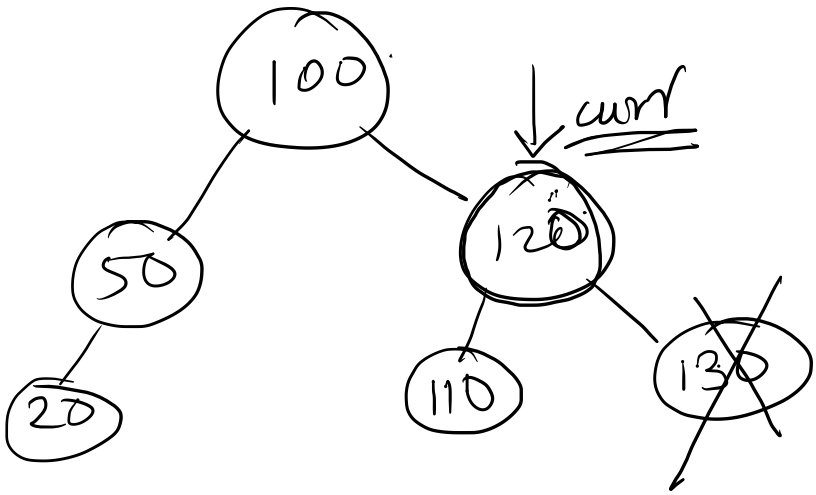
right min



1. Find right min
2. Delete right min
3. curr.val ← min.val

left max

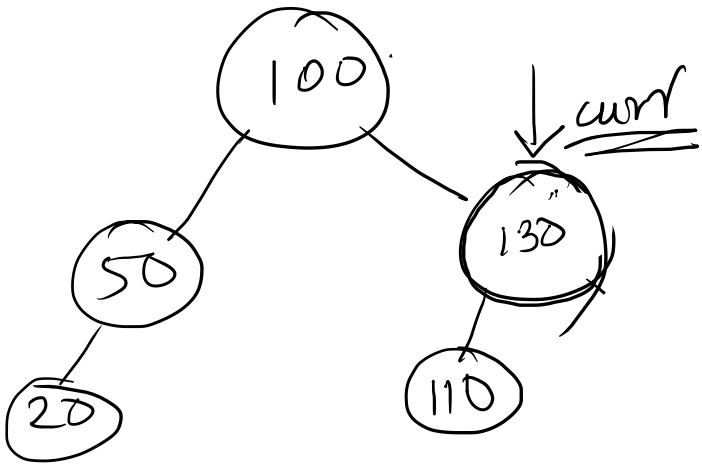
right min



delete 120

(1) rgt min = 130

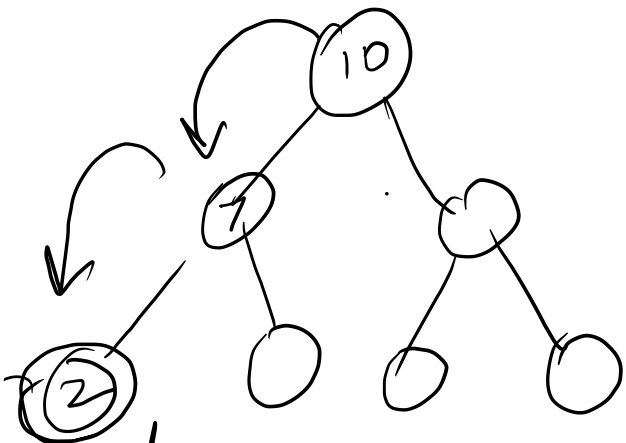
(2) Delete rgt min



delete 120

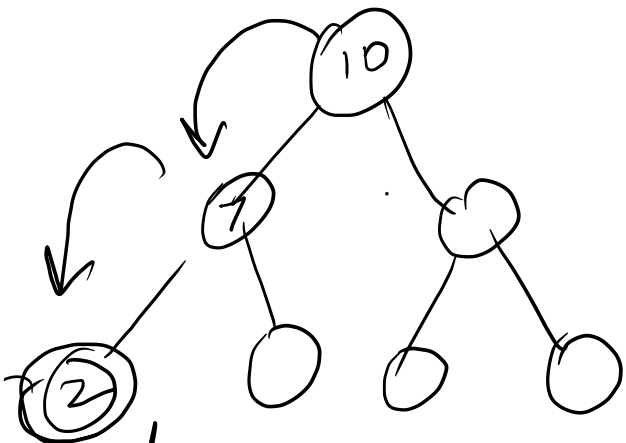
1. rgt min = 130
2. Delete rgt min
3. curr val = min

Extreme left = minimum

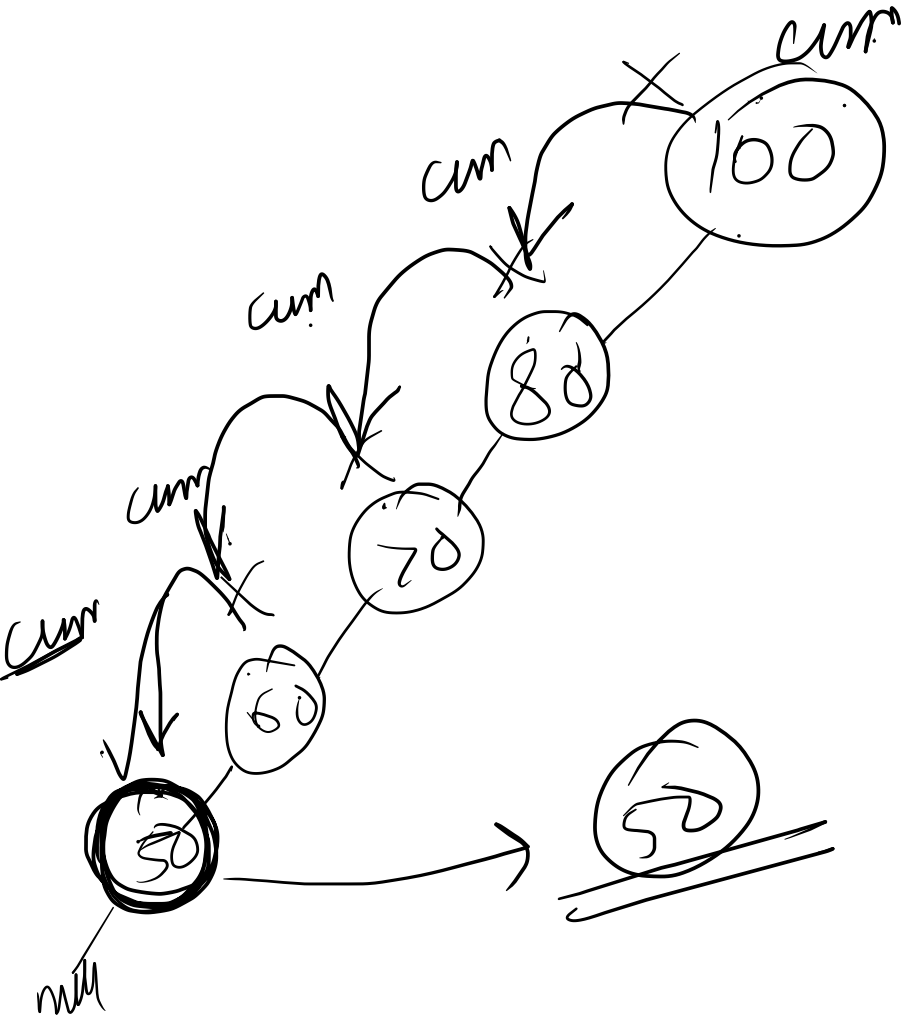


leftmost node

Extreme left = minimum



leftmost node


curr
curr
curr
curr
curr
100
80
70
60
50
50
null