search tree

Lecture 1

2-3 - trees

- What are 2-3 trees??
- Insert / ~~Delete~~ create
- Delete
- Analysis
- Why 2-3 trees

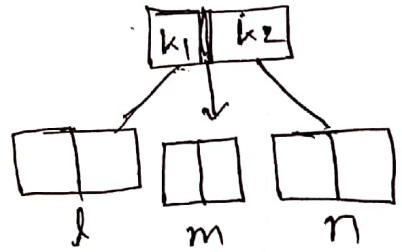
↳ m-way
* Multiway search tree

* Degree 3

height Balanced search tree → * B-tree

* All leaf nodes at same level

* Every node must have $\left\lceil \frac{n}{2} \right\rceil = \left\lceil \frac{3}{2} \right\rceil = 2$
  → cill value
  → children



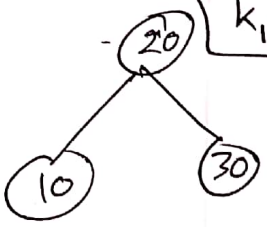
condition value

$k_1 < k_2$

$l < k_1$

$n > k_2$

$k_1 < m < k_2$

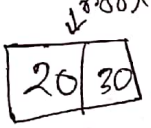


Binary search tree

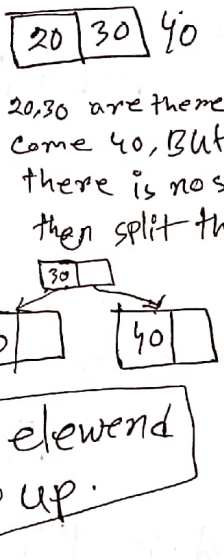
keys → 20, 30, 40, 50, 60, 10, 15, 70, 80

* Insert and see How height is Balanced?

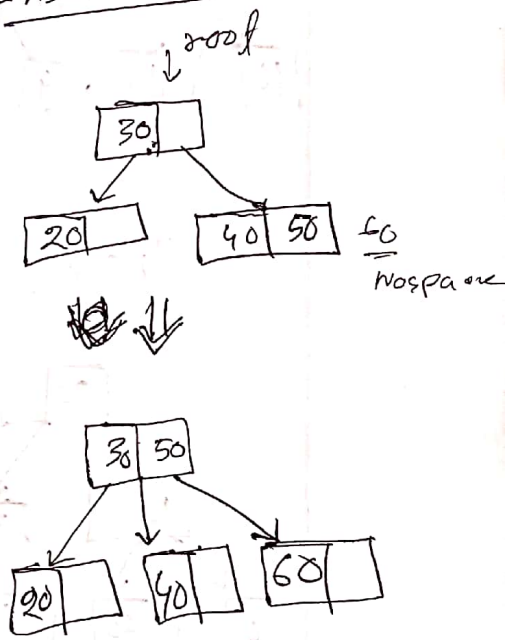
Insert 20, 30



Insert 40



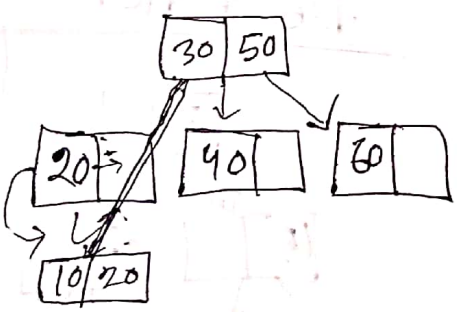
Inser 50, 60



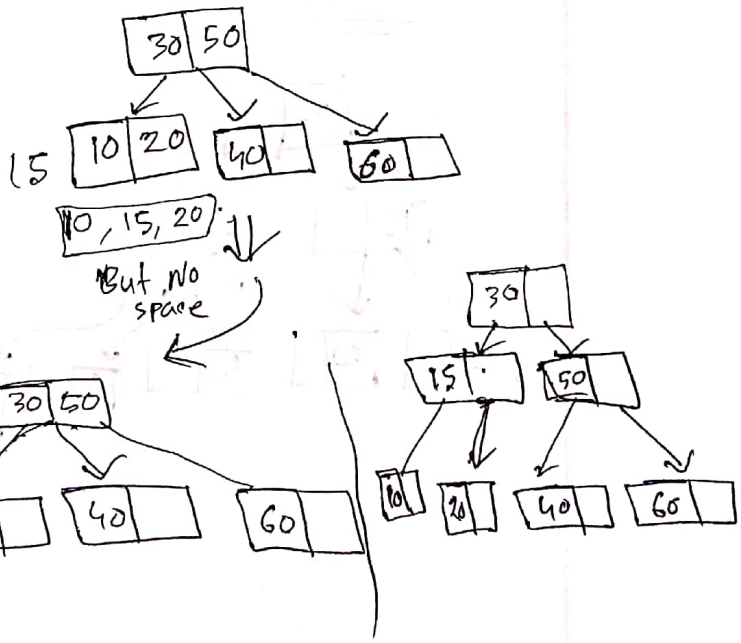
Binary search tree we add elewen nodes in down word

But 2-3 tree we add up words

Inser 10

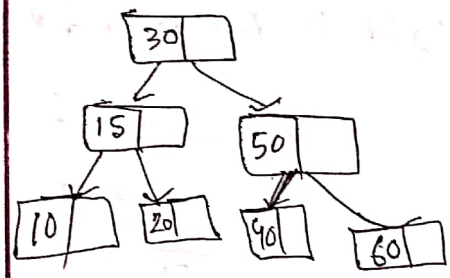


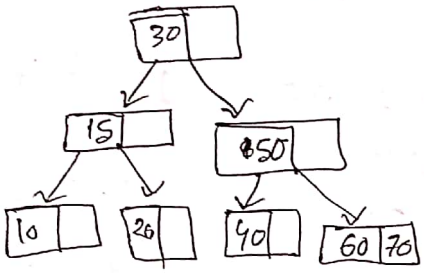
Inser 15



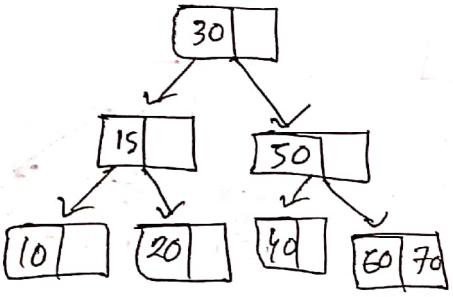
Deletion

Insert 70

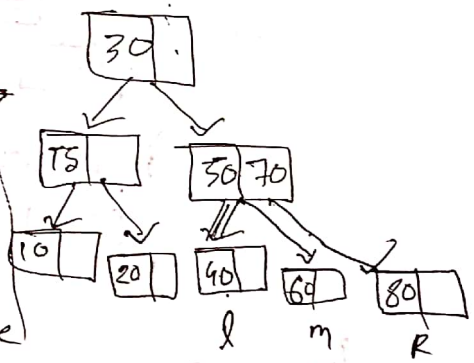




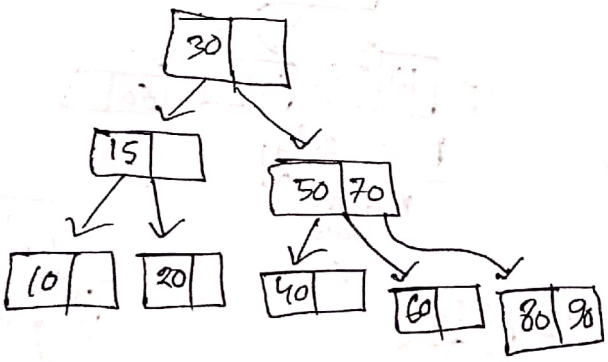
Insert 80



No Space

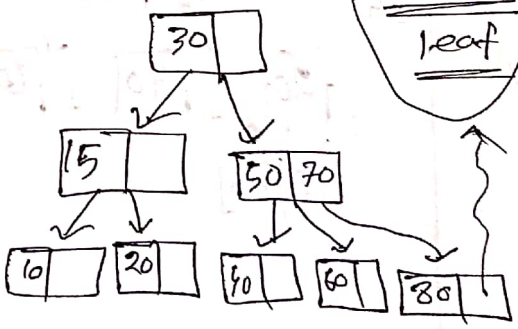


Insert 90



Deletion

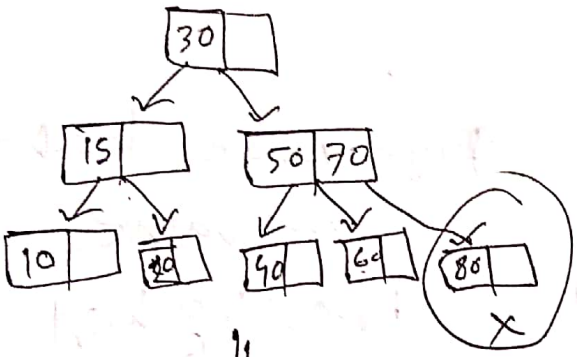
90 case 1 leaf



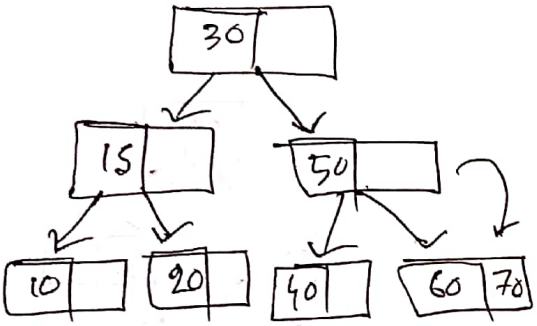
simply Delete 90 from leaf node

case: 1 ⇒ 90
↳ simply delete

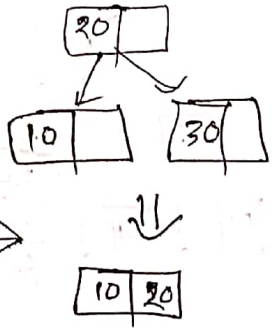
case: 2 ⇒ 80
↳ delet and merge



and join previous value



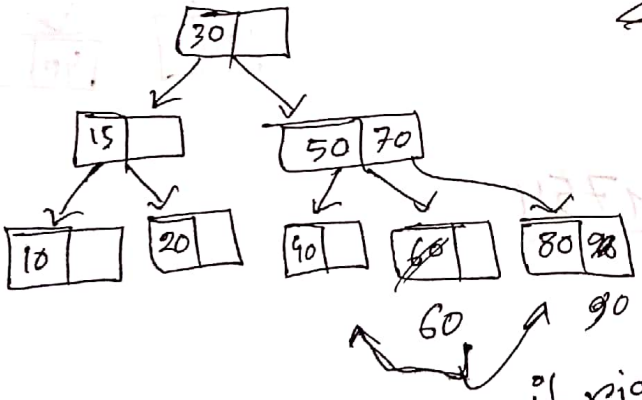
example →



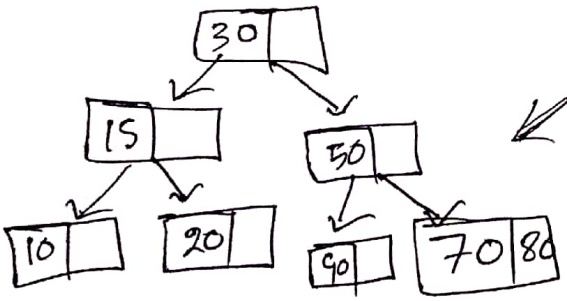
supposed 30 deleted

back again

case: 2 → 60



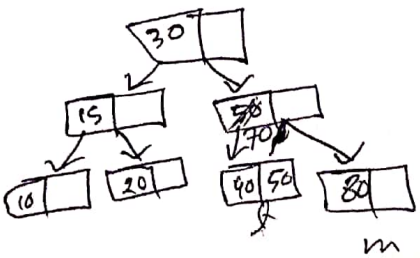
if right



Find 60

when we find this Node is blank we can join it either Left or Right

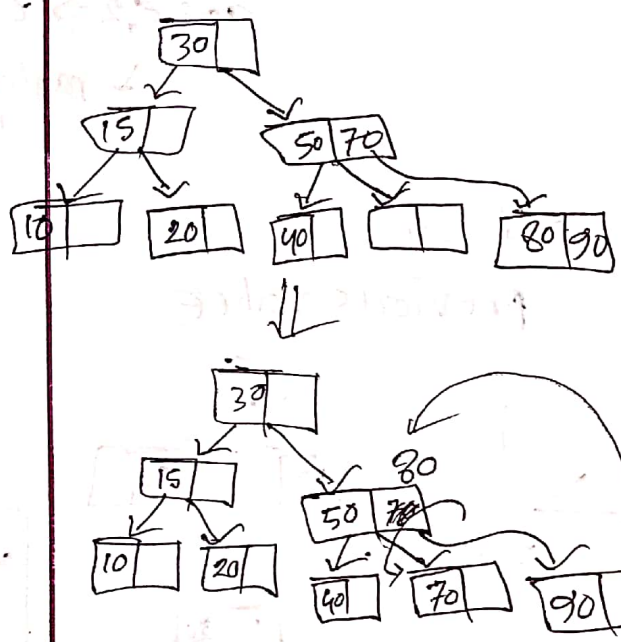
Left sibling



m

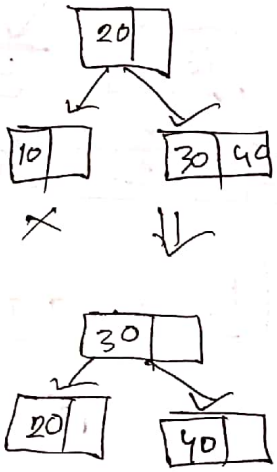
case 3: 60

share key



Right node have 80, 90 two value

browing key vaiya parent



6|7|3|7|0|7|4|7|5|4