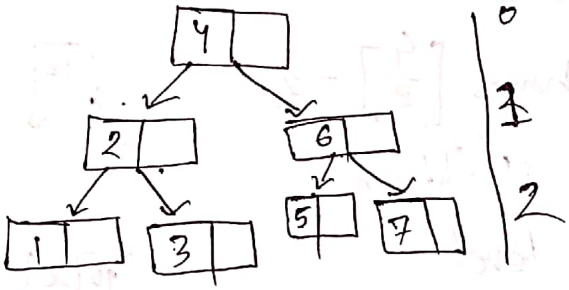
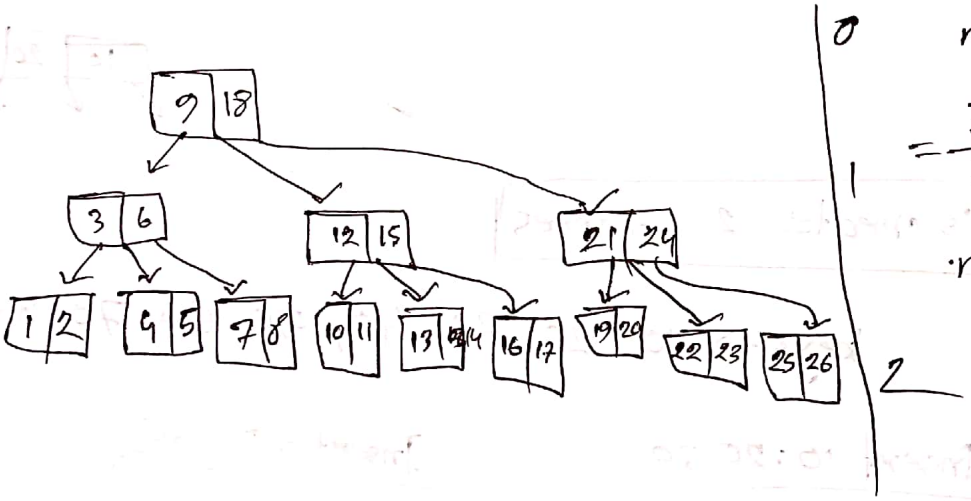
2-3 trees

min heigh and max heigh and Nodes



$$\min n = 1 + 2 + 2^2 + 2^3 \ldots$$
$$= 2^{h+1} - 1 \qquad \left[\tfrac{h}{2}\right] = 2$$

$$\max h = \log_2 (n+1) - 1$$



$$\max n = 1 + 3 + 3^2 + 3^3 \ldots$$
$$= \frac{3^{h+1} - 1}{3 - 1}$$

$$\min h = \log_3 \left[n(3-1) + 1 \right]$$

Using B-trees

Databases

2-3-4 trees

Lecture-2

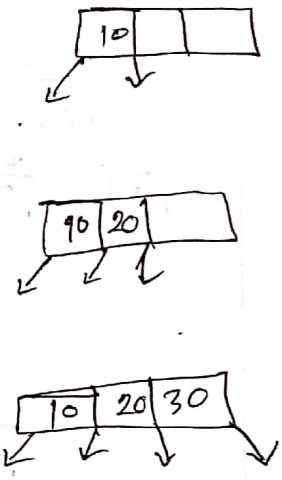
Height Balance multiway search tree.

① B-Tree of degree = 4

② Every Nodes must have $\left\lceil \frac{4}{2} \right\rceil = 2$ children

③ All leaf at same level

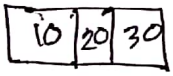
d = 4



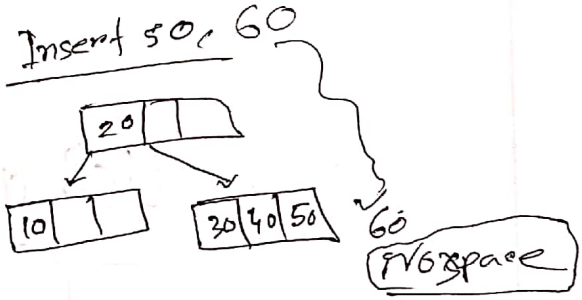
Let's create 2-3-4 trees

keys → 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110

Insert 10, 20, 30

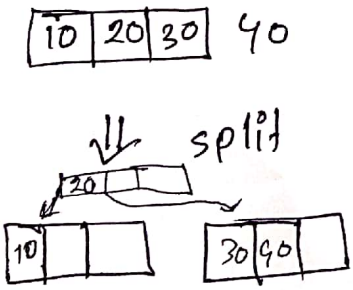


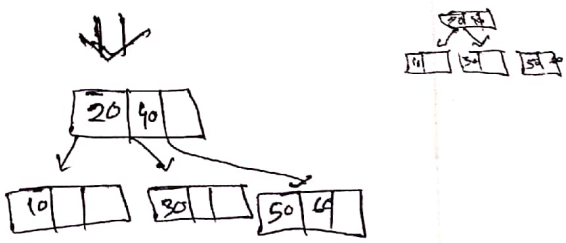
Insert 50, 60



NEW

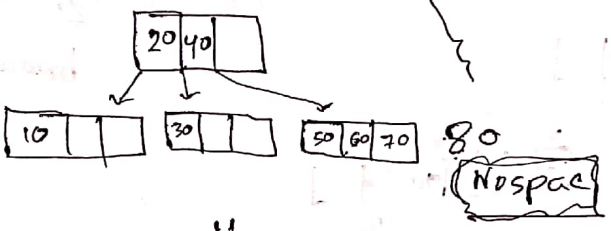
Insert 40,



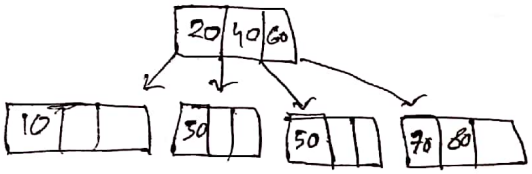


When Even number of key then consider Left bias / Right bias

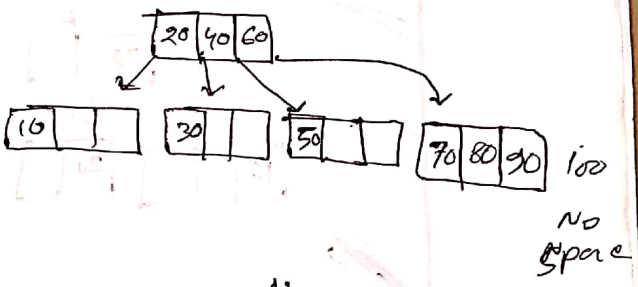
Insert 70, 80



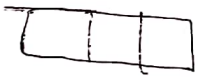
⇓ 80

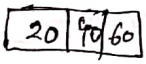


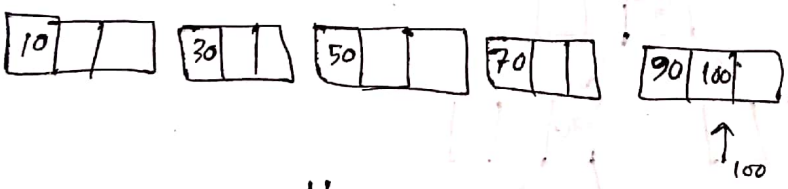
Insert 90, 100



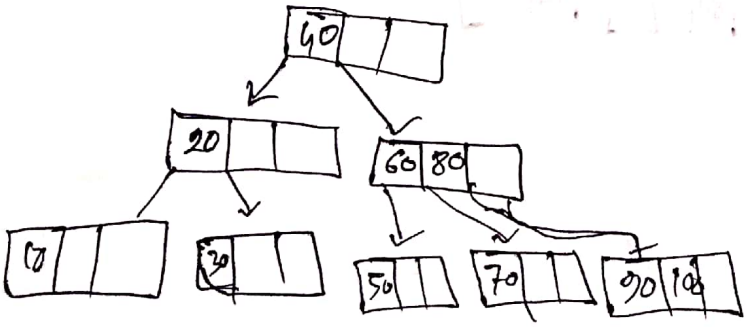
⇓ 100







⇓

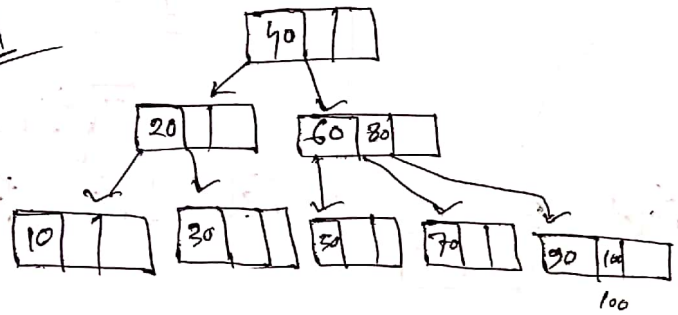


How to delet from 2-3-4 trees

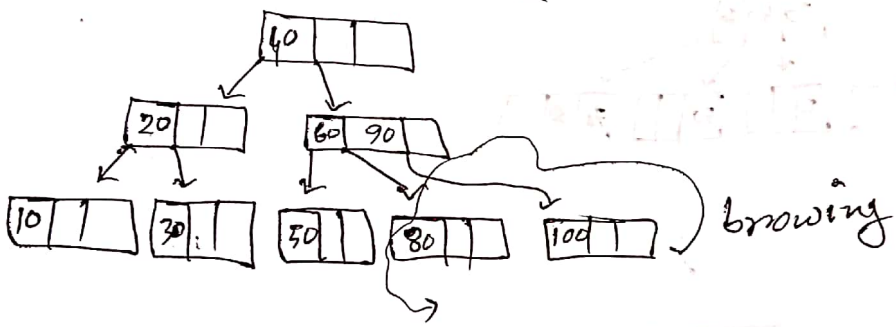
main

① case: 100

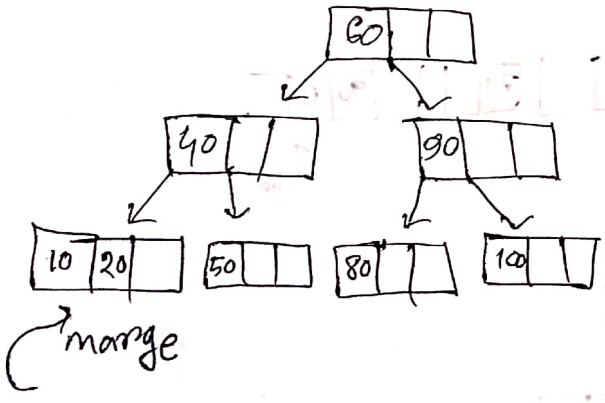
② case: 70
browing key



70



30 marge

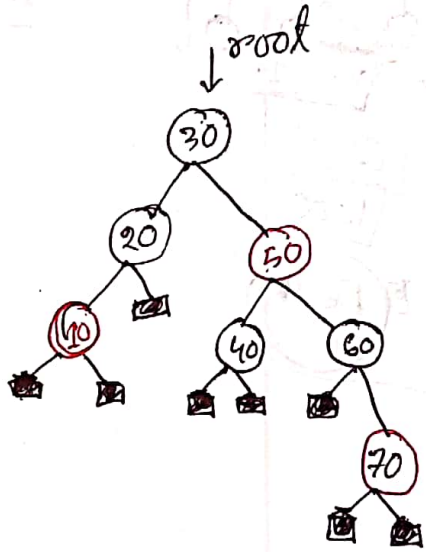


Red - Black - tree

Leture 3

1. Its a Height blanced Binary search tree, similar to 2-3-4 trees
2. Every Node is either Red or Black ⇒ O O
3. ** Root of a tree is Black
4. Null is also Black
5. Number of Black on path from root to leaf are same (3)
6. No 2 consequtive Red and Parent and children of Red are black
7. New Insert Node is Red → O
8. Height in $\log n \le h \le 2\log n$

like →



9+3=12

40

Lecture 4

Red-Black tree

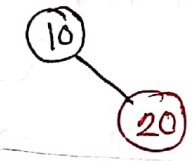
Create

Keys : 10, 20, 30, 50, 40, 60, 70, 80, 48

Insert 10

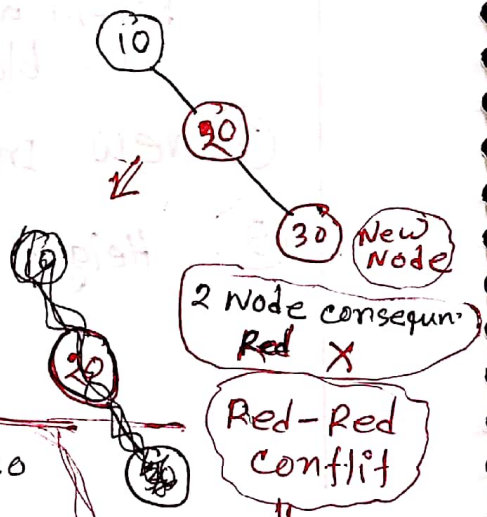
10 ⇒ [root Node Always Black] ⇒ 10

Inser 20



New Node always Black
it is parents Black
No problem

Inser 30

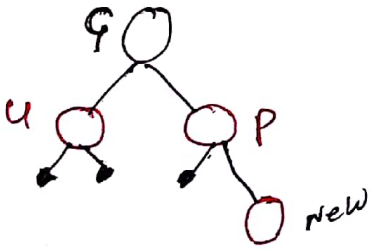


2 Node consequn. Red X

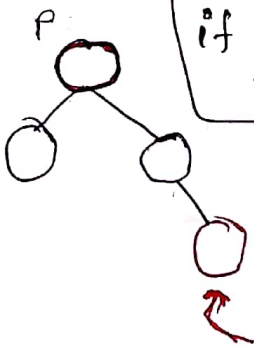
Red-Red Conflit

① Recolor
② Rotation

① color
New Node is Red it's parent also Red then check
Uncle is Red



color ⇒

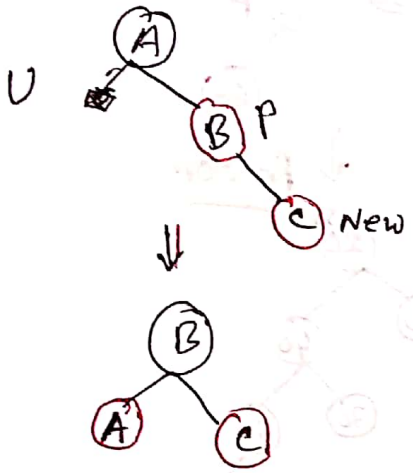


if Root P the it Black

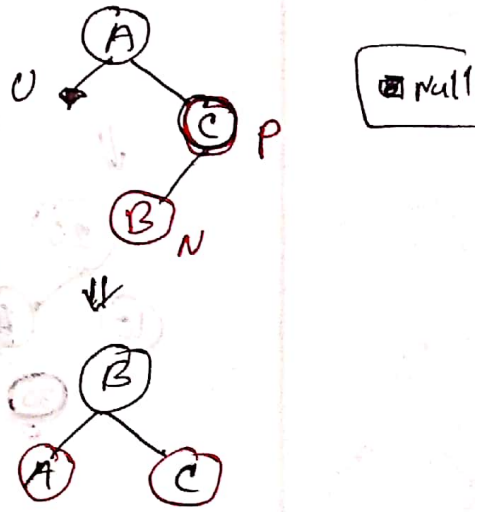
same as it is

Uncle is Black Rotation

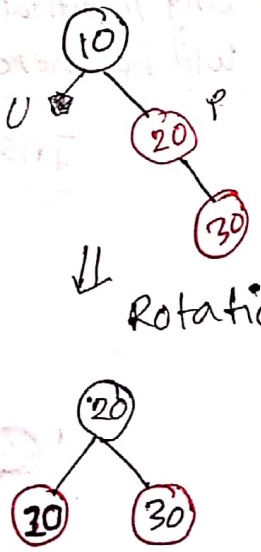
Zig-Zig LL/RR



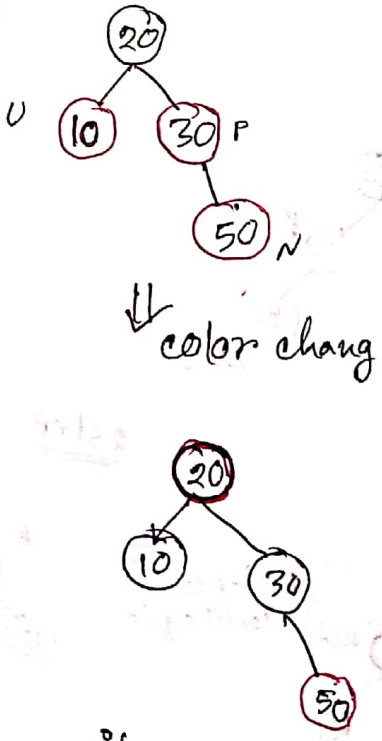
Zig-Zag RL/LR



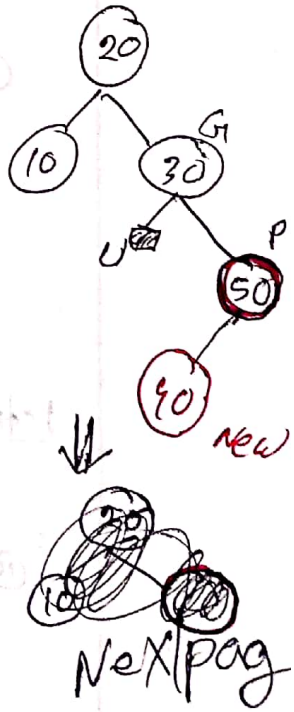
Inser 30



Inser 50

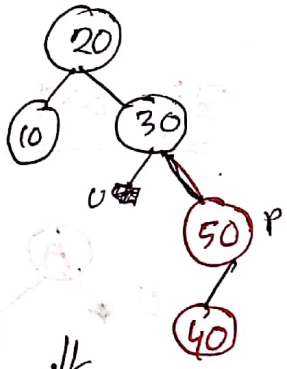


Insert 40

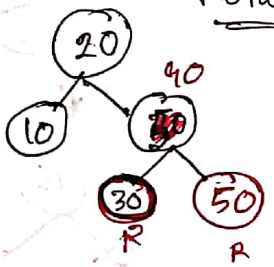


if G parent is Root then mak it black

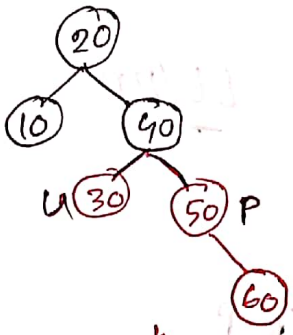
Insert 40



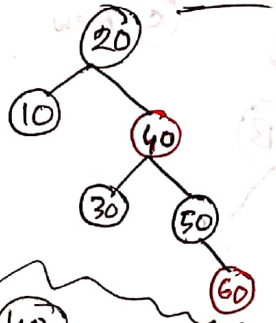
Rotation



Insert 60

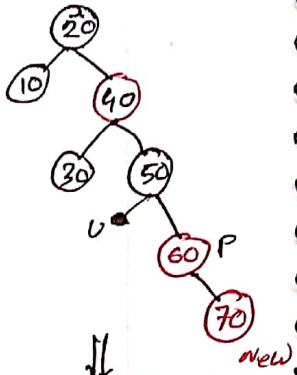


Recolor

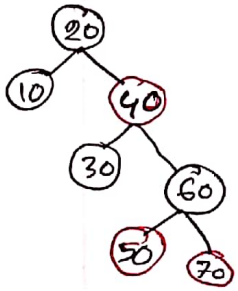


40 → Node upper Part also check and if any modification then will be there

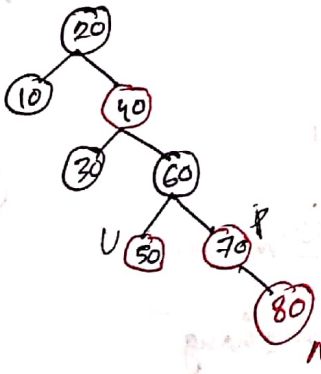
Insert 70



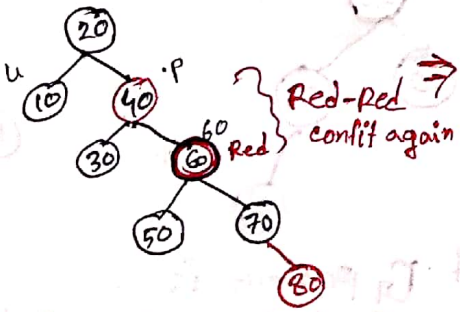
Rotation (RR)



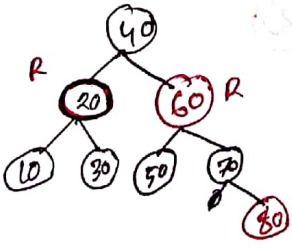
Insert 80



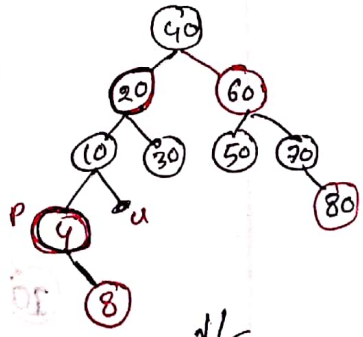
1 step

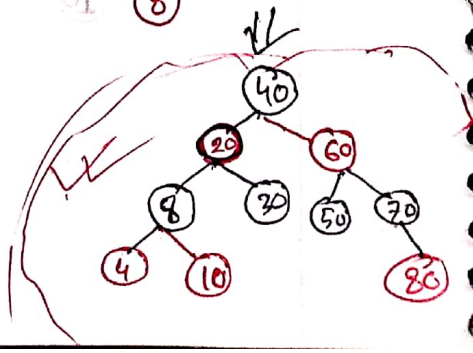


2 step



Insert 4, then 8

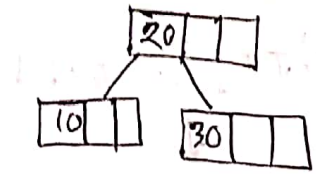


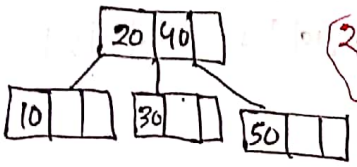


Red-Black tree

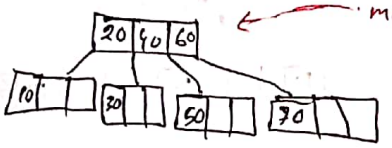
Lecture 5 — 2-3-4 tree and Red-Black tree

2-3-4 tree



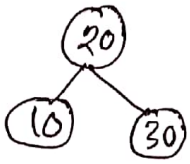


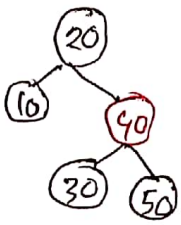
20, 40 in same node

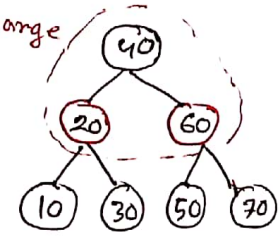


merge

Red-Black tree

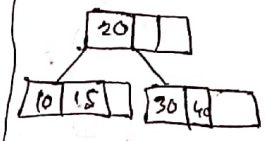




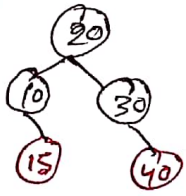


Example

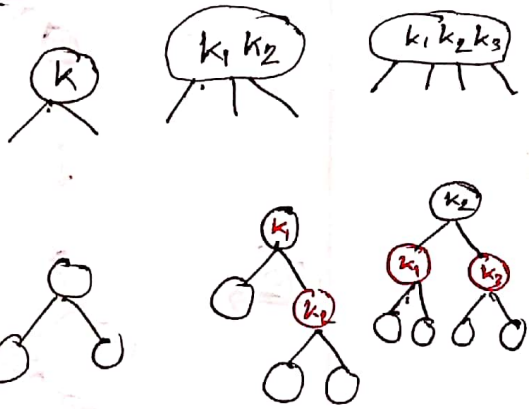
2-3-4 tree



Red-Black



General formula:

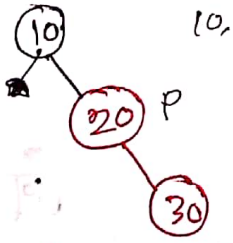


two children black color

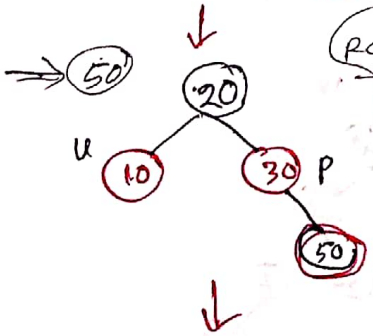
Lecture 6

keys: 10, 20, 30, 50, 40, 60, 70, 80, 4, 8

Red-Black tree



10, 20, 30



rotation 30

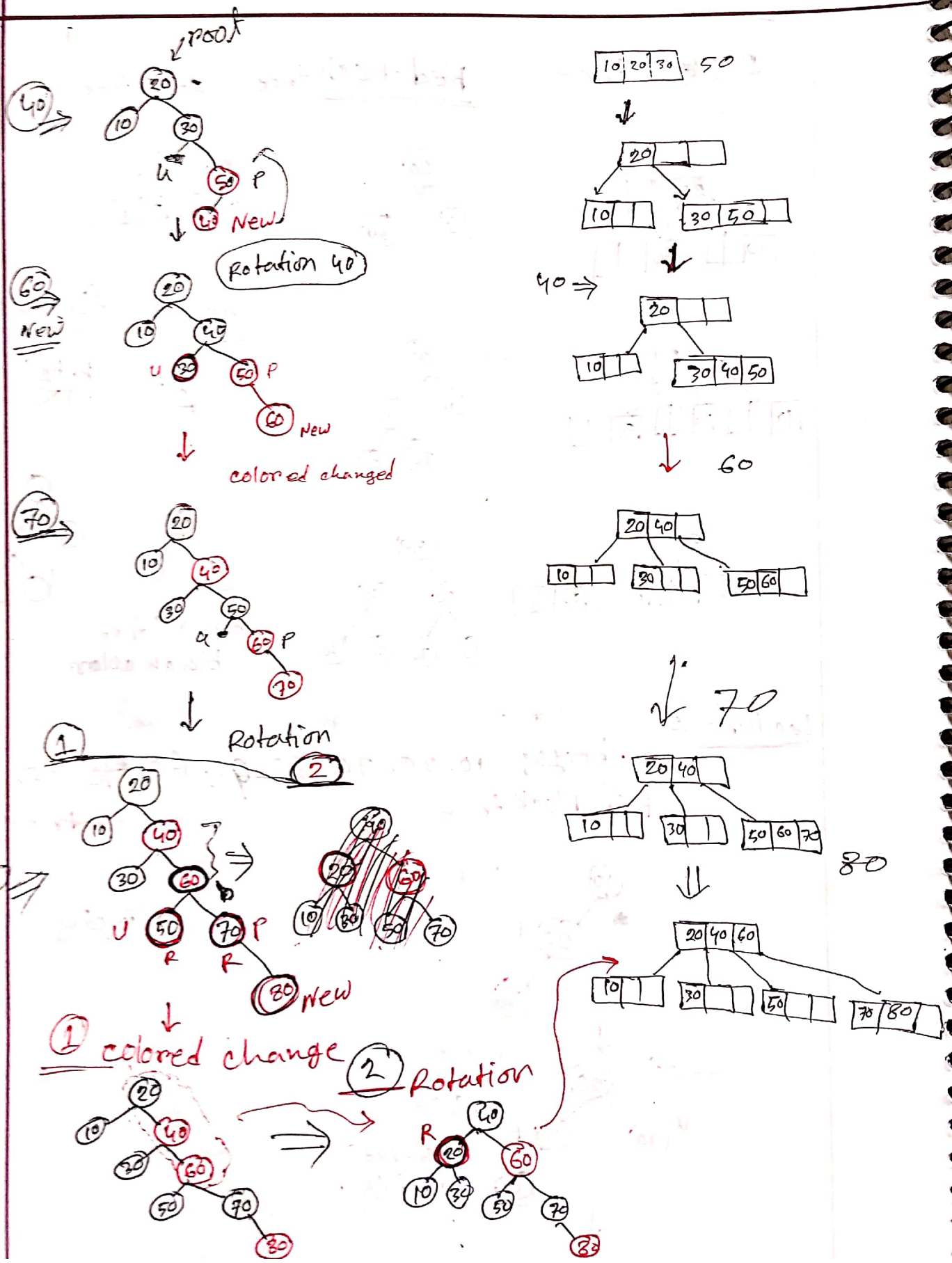
10	20	30

Red-Red conflict

2-3-4 trees

10	20	30

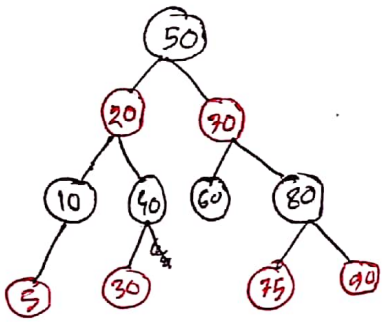
Keys : 10, 20, 30, 50, 40, 60, 70, 80, 4, 8



Deletion from Red-Black tree

Lecture 7

Red-Black-tree ~~and~~ Deletion similar Binary search tree



90 ⇒ simply Delete coz leaf Node

20 ⇒ * We don't delete the Node
* we delete the value
* who will take it's place
inoder predecer or inorder successer

10 30 40

Inorder: 5, 10, 20, 30, 40, 50, 60, 70, 75, 80, 90

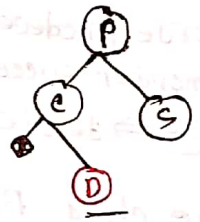
take place 20

* when ever deleted A Node from BST then leaf Node will bedeleted or who ever only on child.

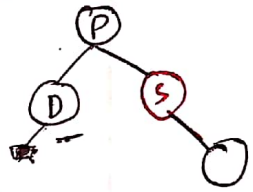
case 1

when Red Node and don't have any children delet it

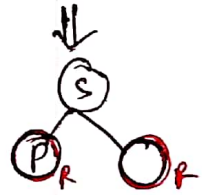
~~case 2~~



case 2

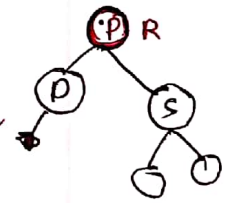


when Deleted Node is Black the check sibling color it is Red then perform Rotation

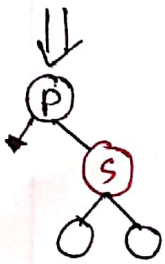


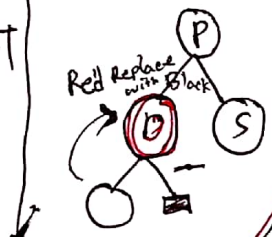
case 3

1



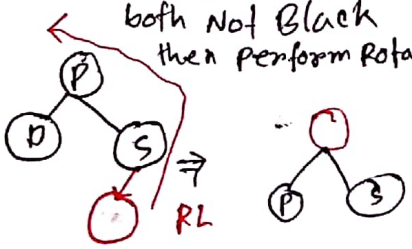
whe delete Black Node and its siblings also Black (S) then check the siblings childre both the color is black then change the color.





2

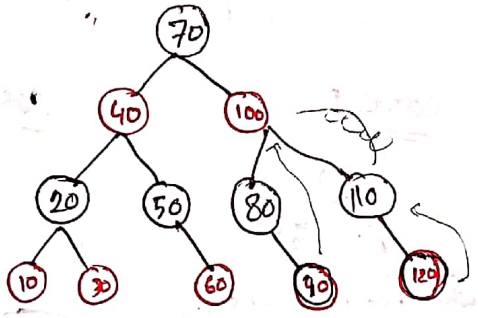
siblings color both Not Black then perform Rotation



Deletion

Leture 8

7x15
105



90 ⇒ simply deleted

100 ⇒ Find inorde predecer
or inorder successer
(90) ⇒ delete (90) node

110 ⇒ have one child Bring 120
the color Black then
deleter (120) Red node

Red-Black and Deletion

*

2-3-4-trees

Lecture: 9