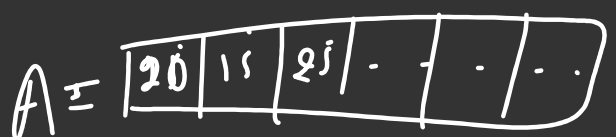
Date
31/02/23

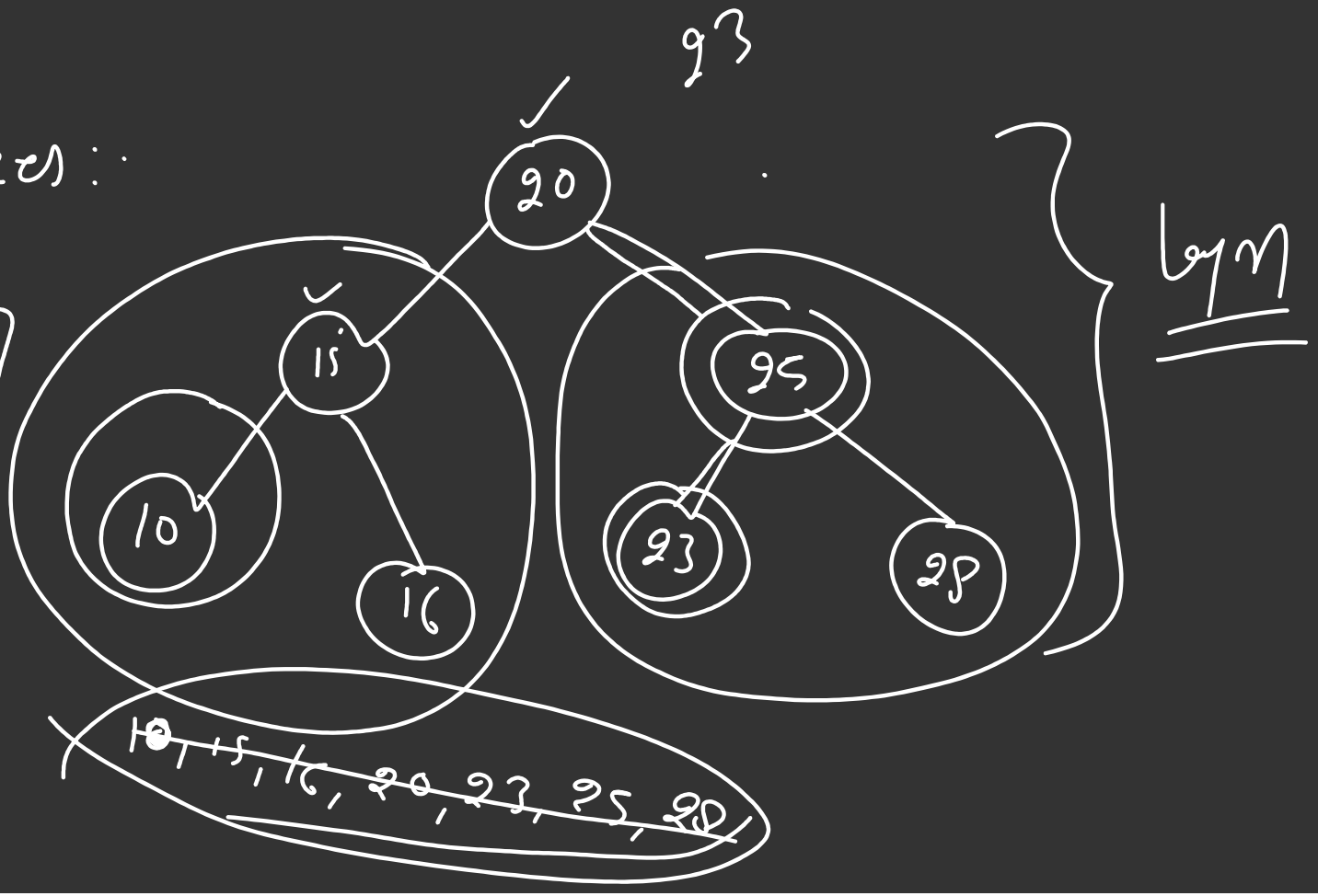
§ Data Structures §

Trees

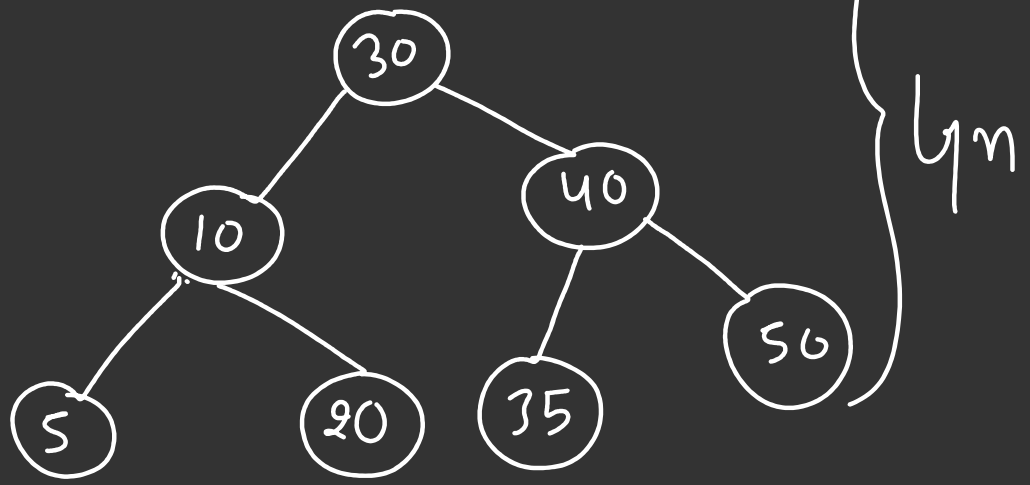
§ Binary Search trees ::



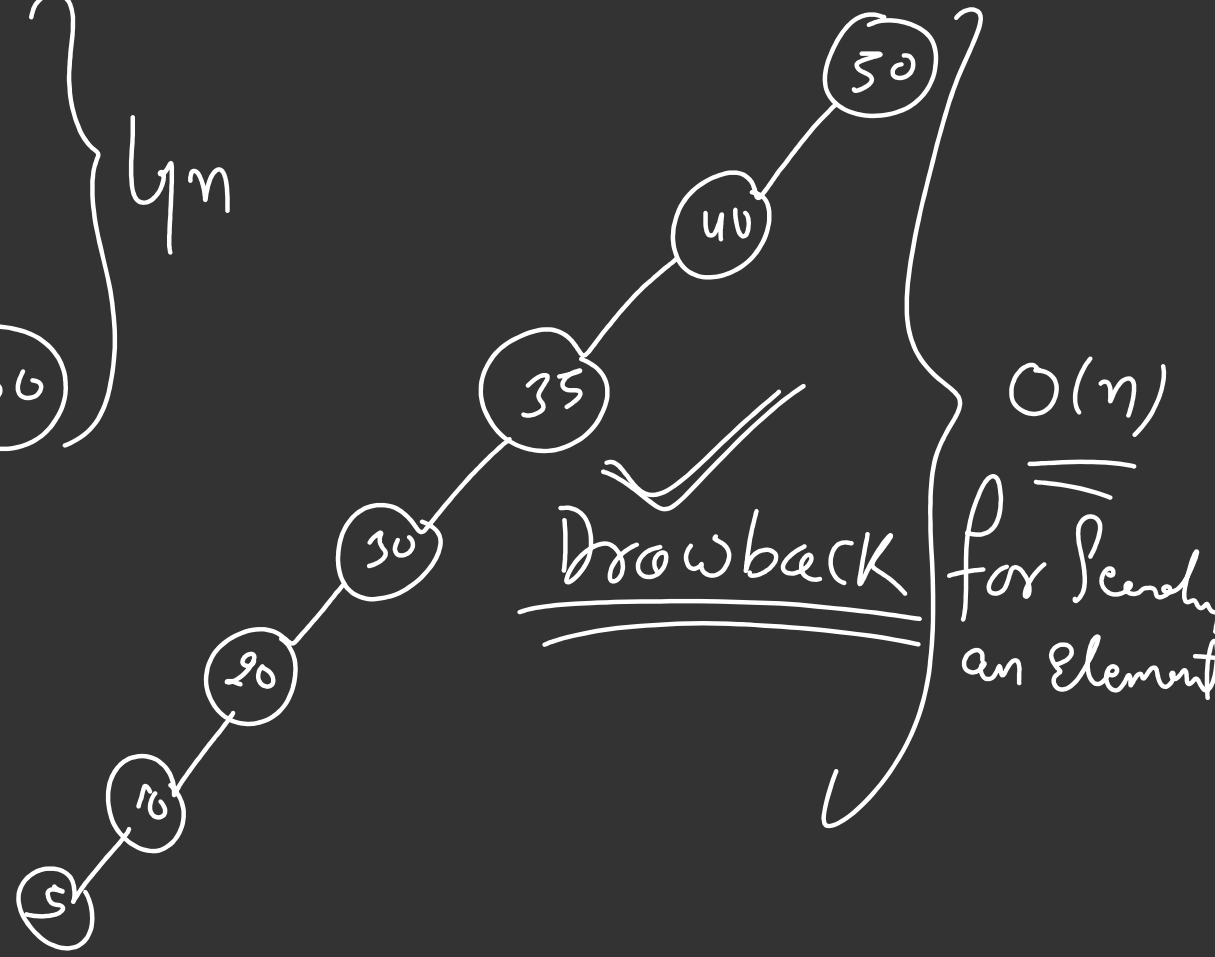
O(n)



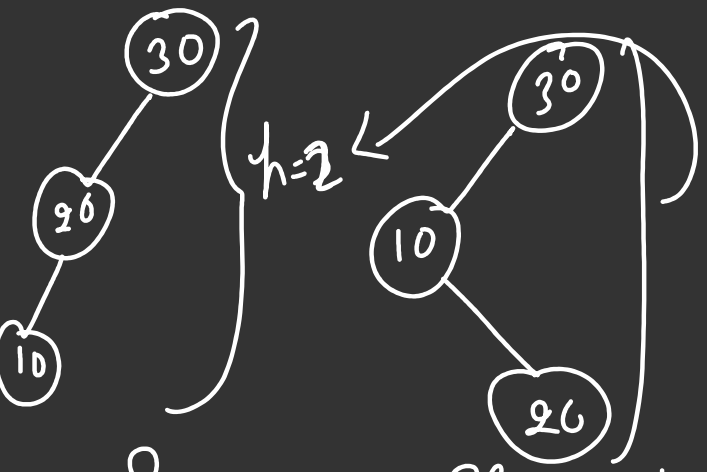
i) 30, 40, 10, 50, 20, 5, 35



ii) 50, 40, 35, 30, 20, 10, 5

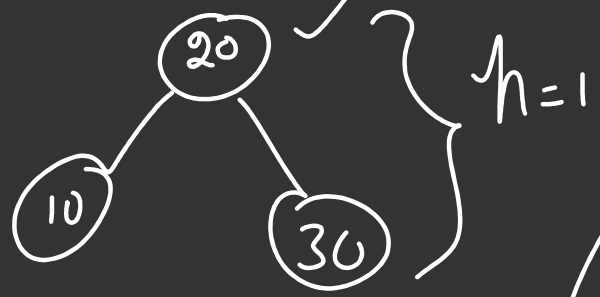


Keys: 30, 20, 10

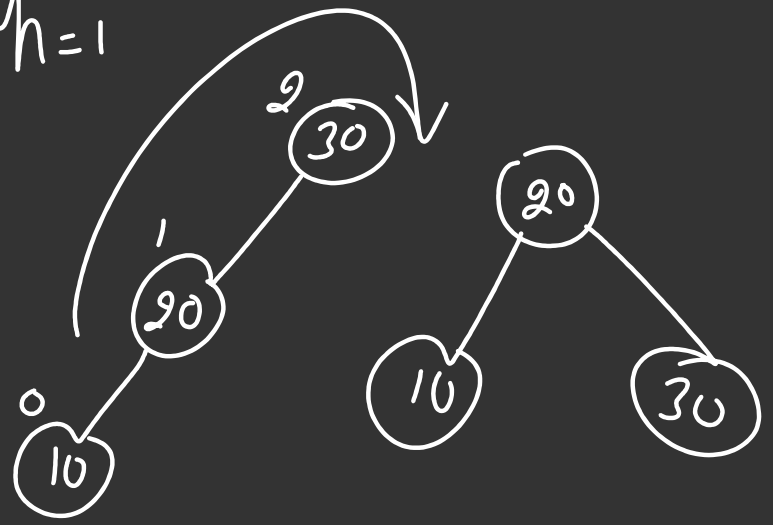


30, 10, 20

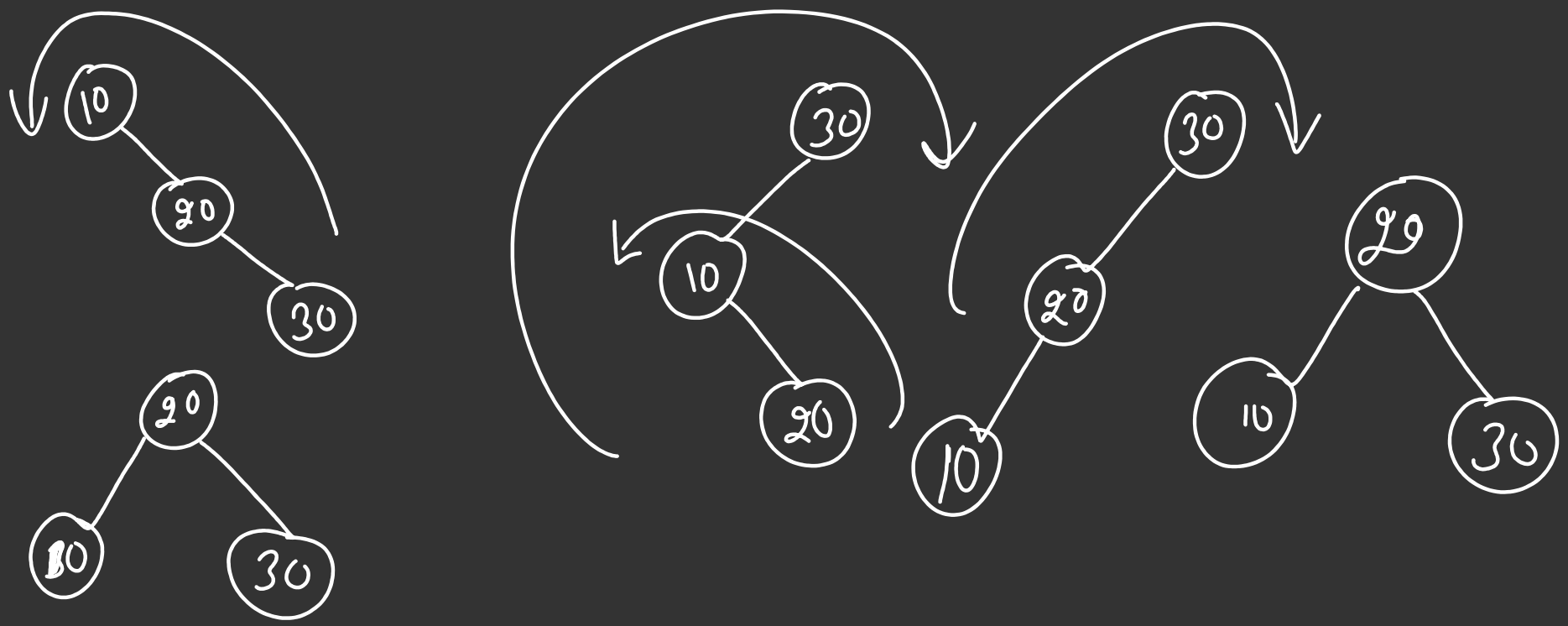
20, 10, 30



AVL tree

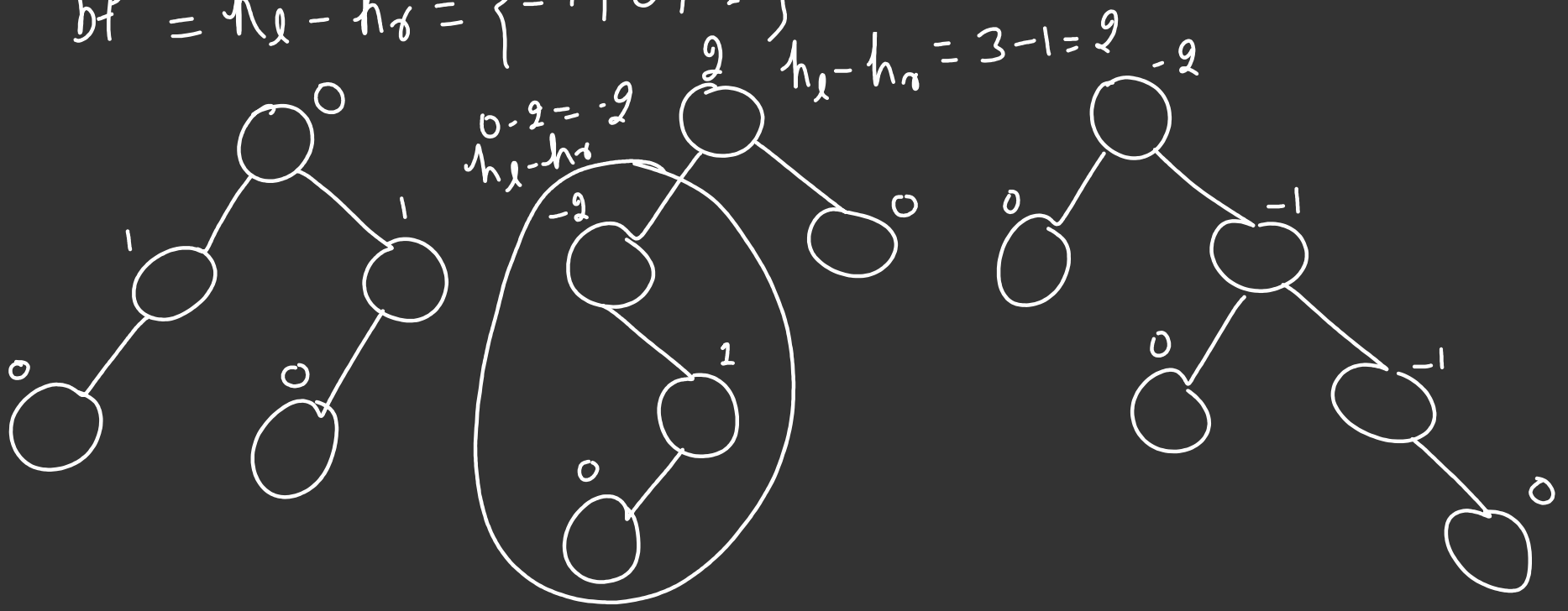


Search an Element in O(log n) time


10
20
30
20
10
30
30
10
20
20
10
30
20
10
30

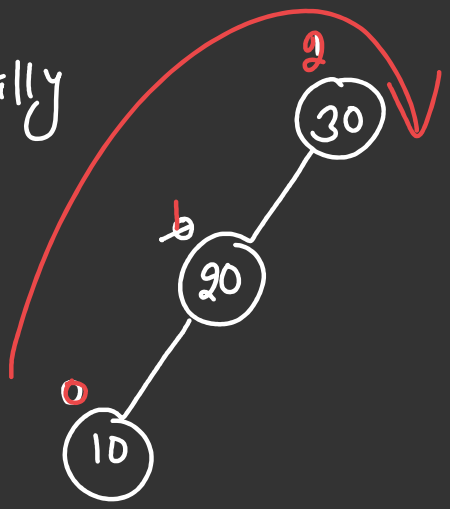
Balance factor = height of left subtree - height of right subtree

$bf = h_l - h_r = \{-1, 0, 1\}$



Now Come to the rotations::

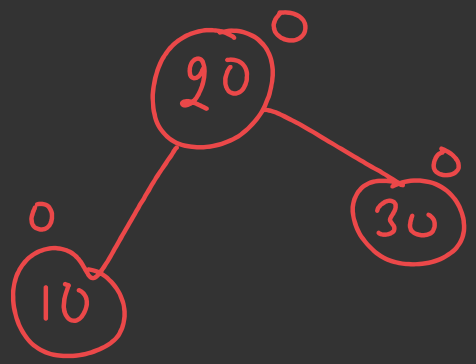
Initially



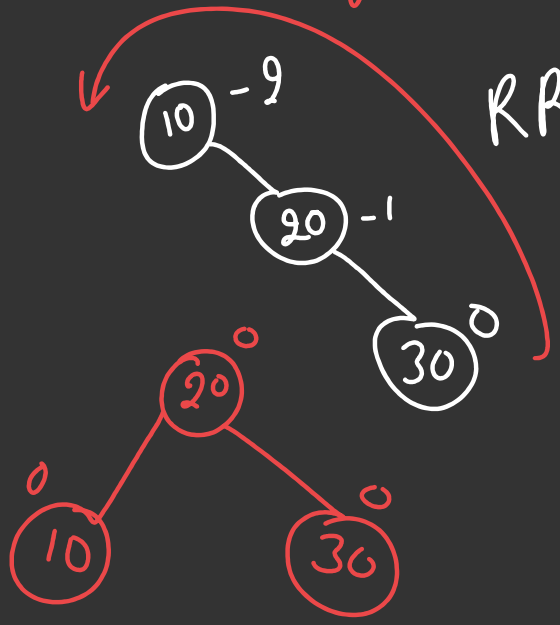
Insert 10

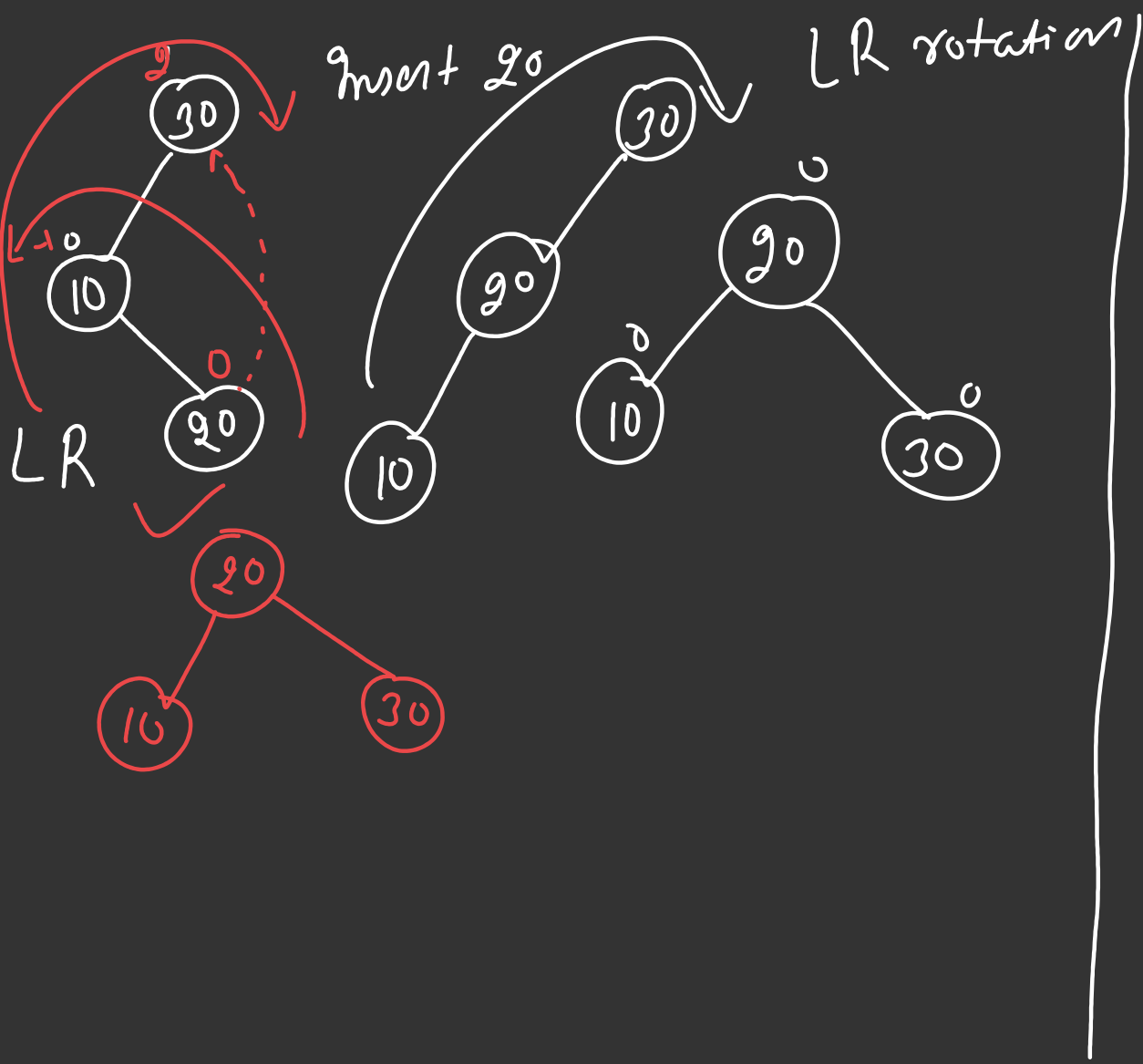
Imbalance due to Left to left Insertion

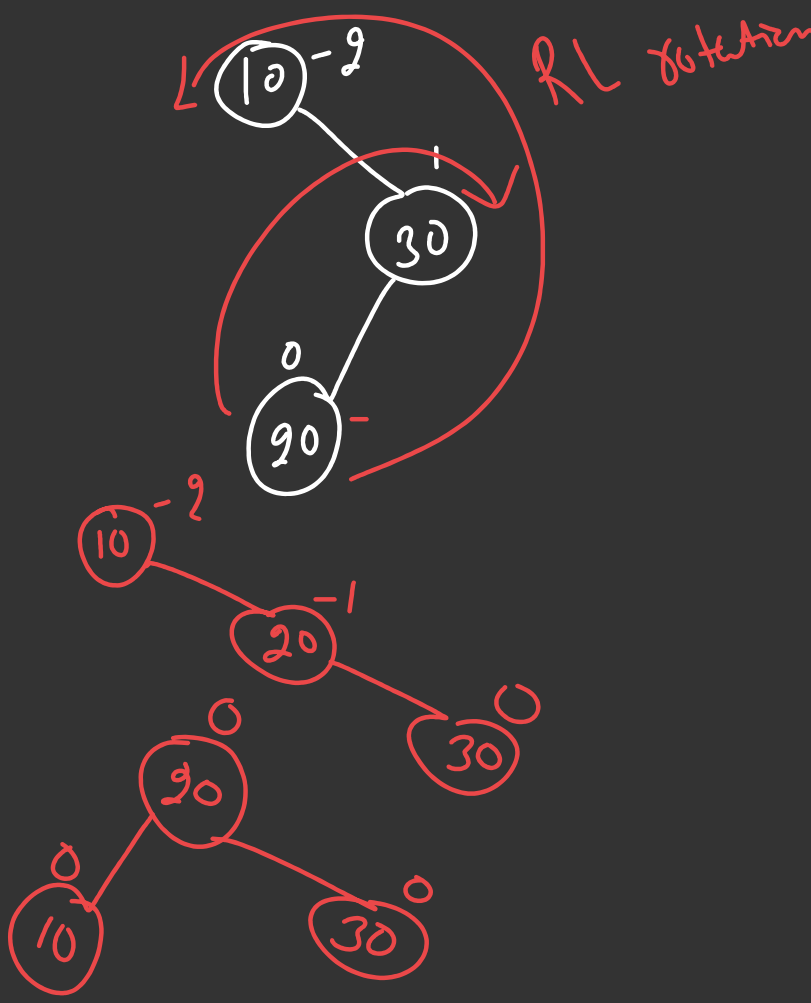
LL rotation

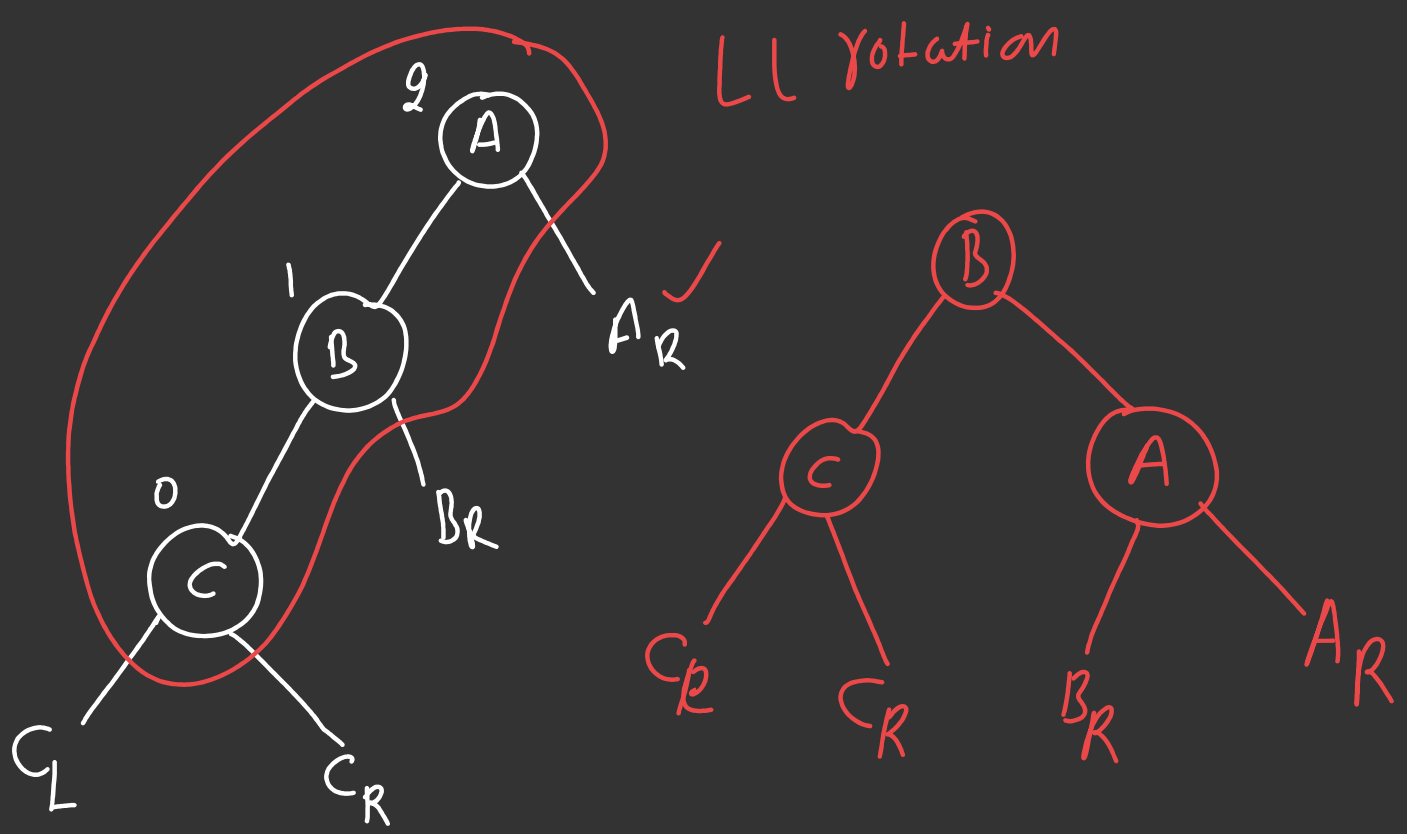


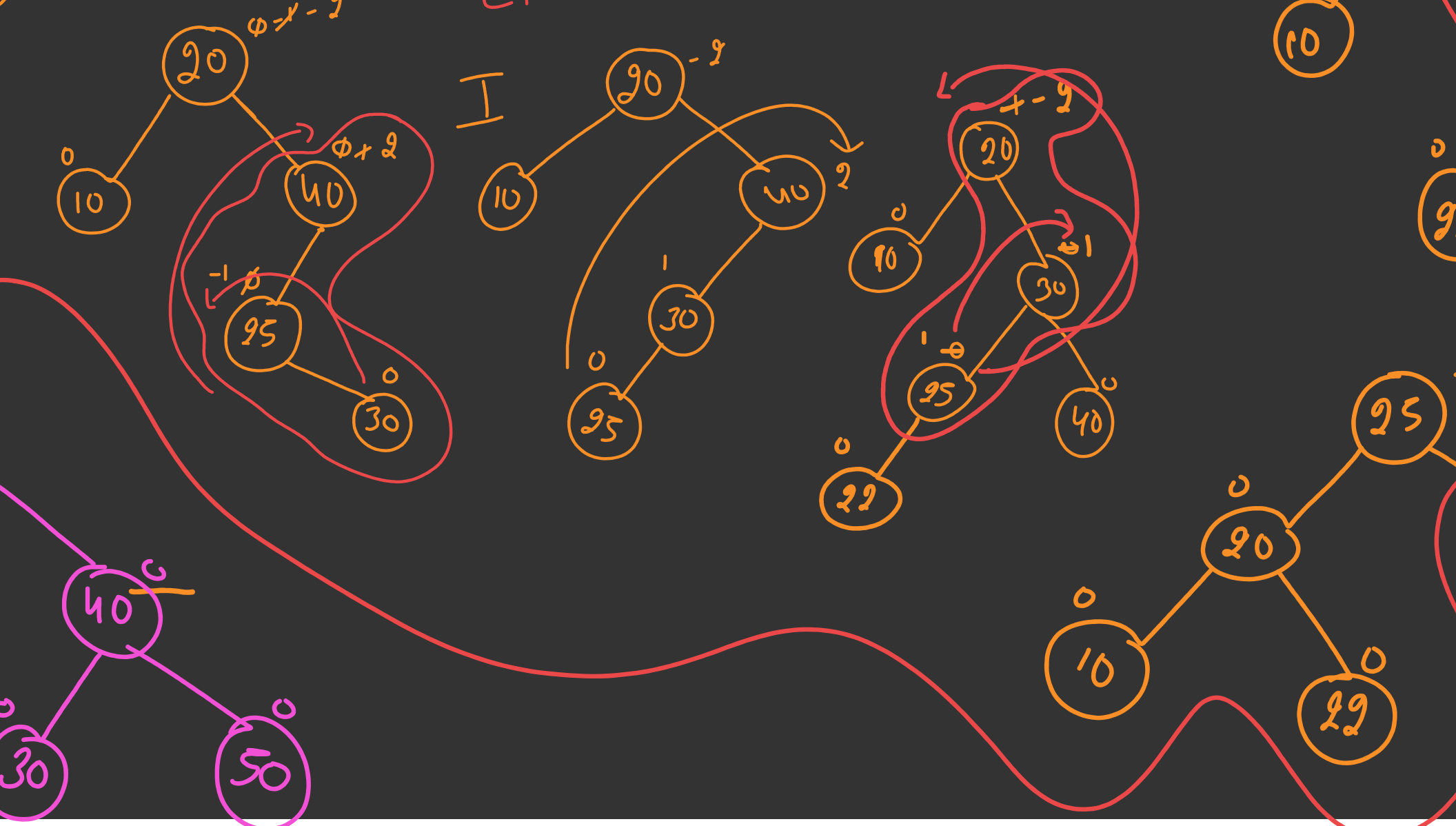
RR rotation

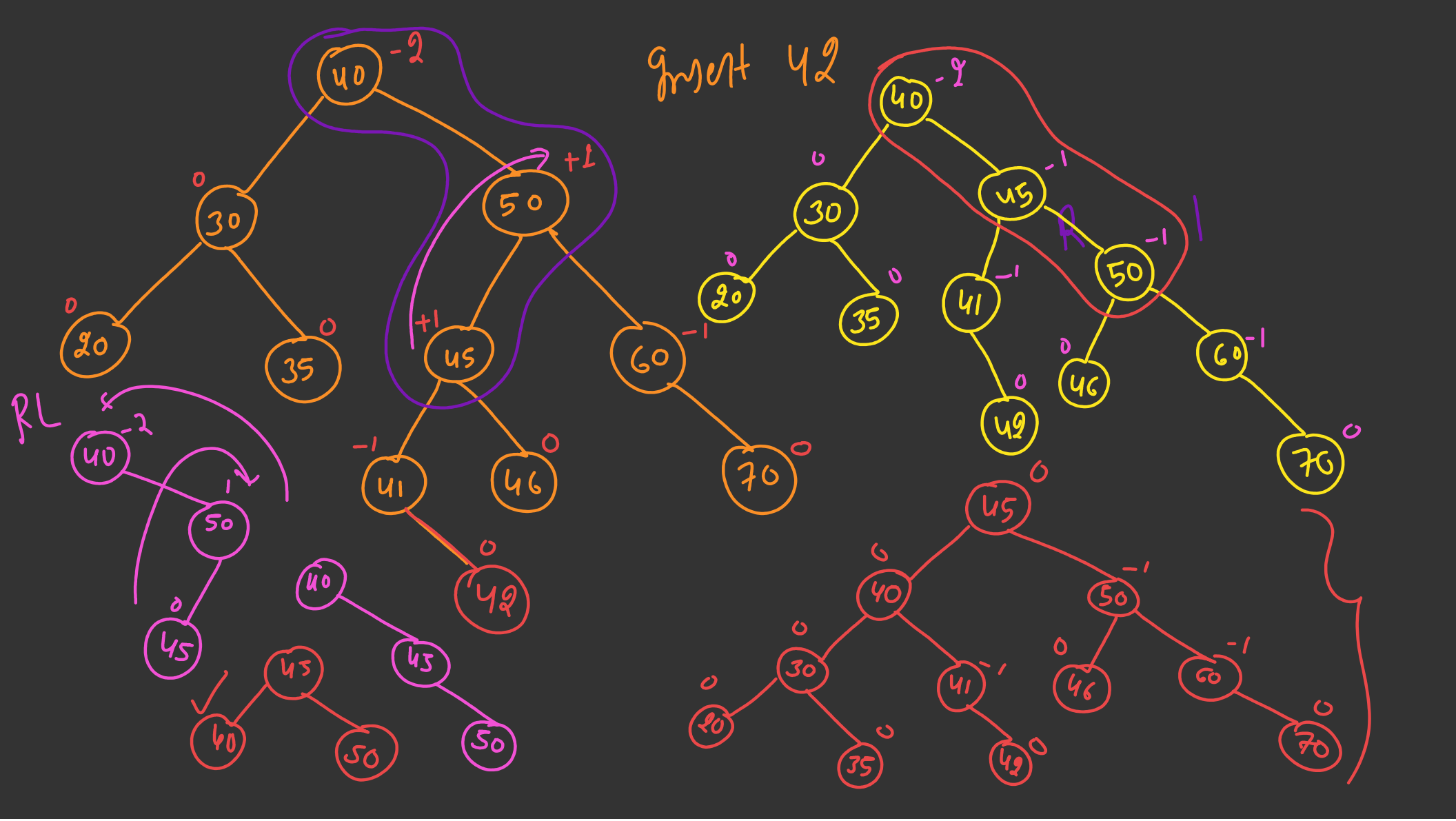



2
30
-1
10
0
90
0
LR
Insert 20
30
90
10
LR rotation
90
0
10
0
30
0


10
-2
RL rotation
1
30
0
90
-
10
-2
20
-1
30
0
90
0
10
0
30
0


LL rotation
2
A
1
B
0
C
A_R
B_R
C_L
C_R
B
C
A
C_L
C_R
B_R
A_R


20
10
40
95
30
90
10
40
95
30
20
10
30
95
40
22
40
30
50
90
10
22
95


insert 42
RL
RR

Assignment:: Jan, feb, mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec.

Create AVL