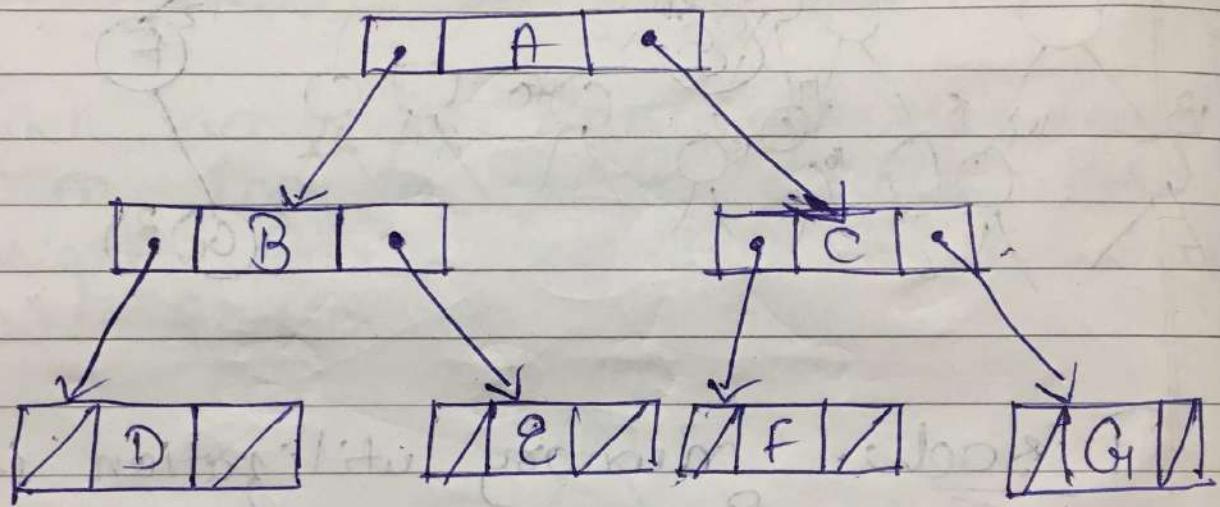
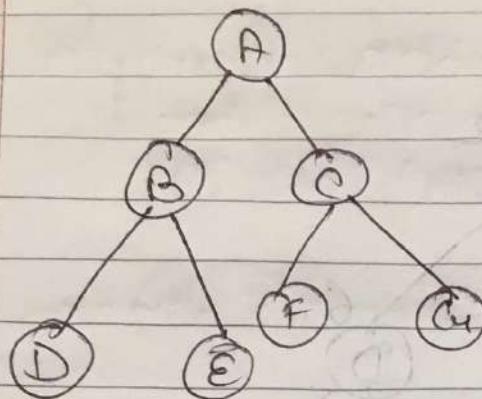
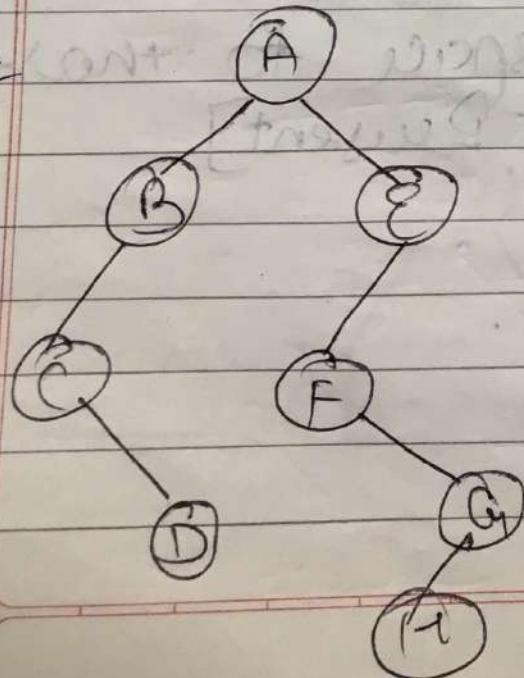


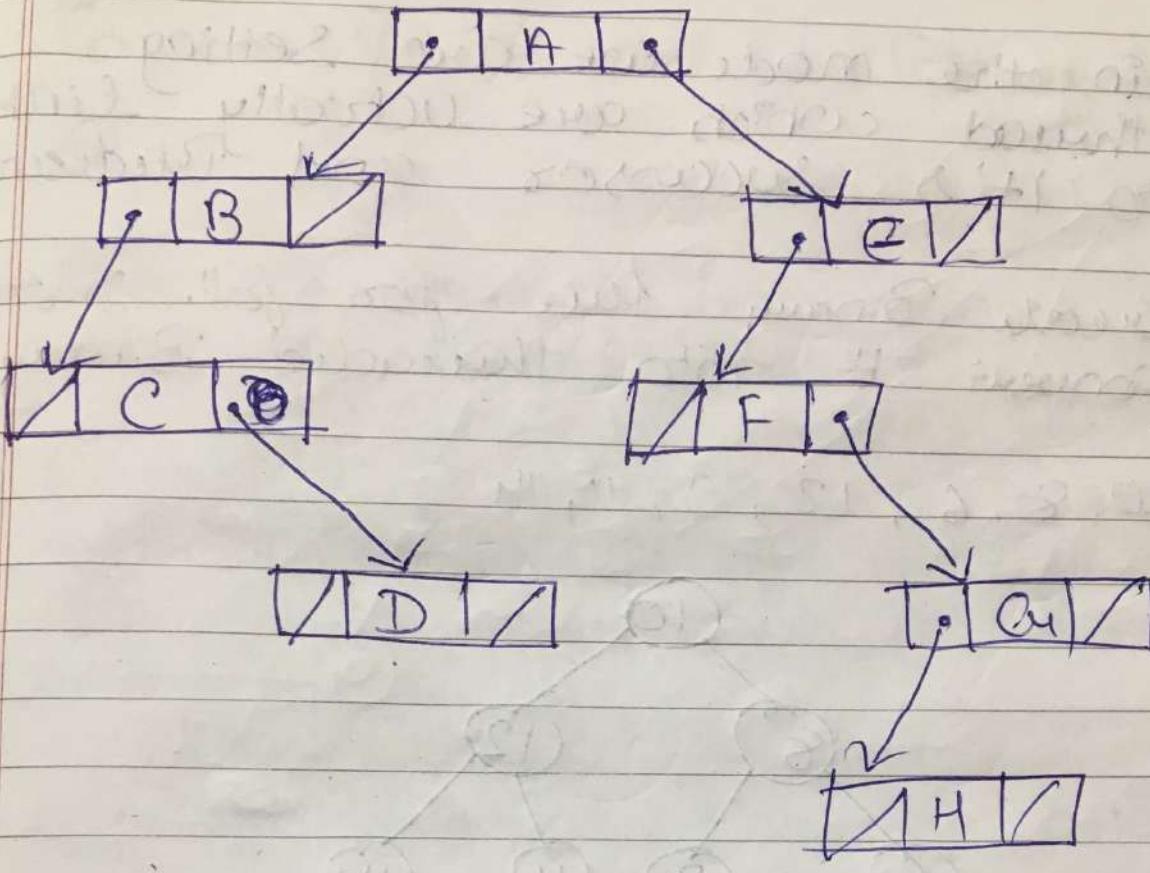
# LINKED REPRESENTATION

Ex:-



Ex:-



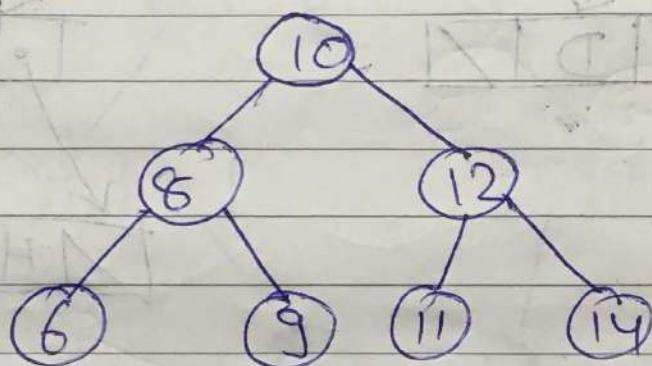


\* Threaded Binary Tree (Inorder threaded B.T.)

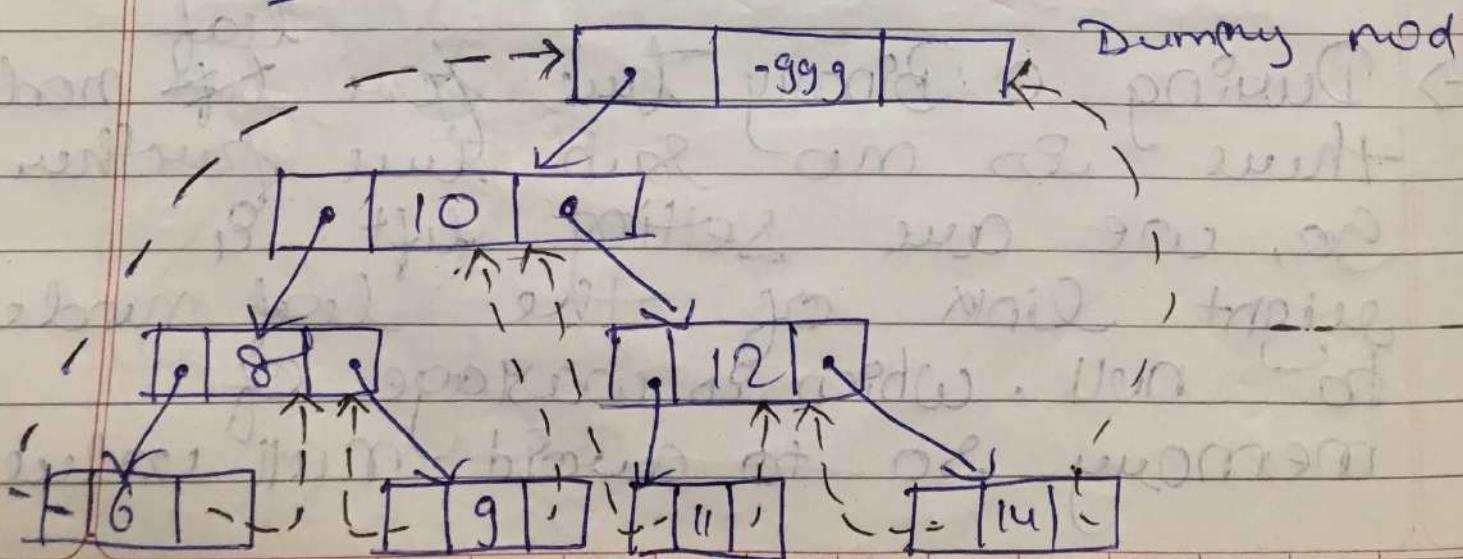
→ During a Binary Tree for ~~left~~<sup>leaf</sup> node there is no sub tree further so, we are setting left & right link of the leaf node to null. which is wastage of memory so to avoid null value

in the node we are setting threads which are actually link to it's successor and predecessor.

- \* Create Binary tree for foll. Data & convert it into threaded Binary tree  
10, 8, 6, 12, 9, 11, 14



Inorder: 6 - 8 - 9 - 10 - 11 - 12 - 14



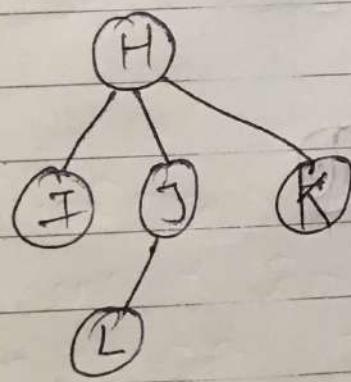
\* Convert General tree to Binary tree

→ Rule for Conversion

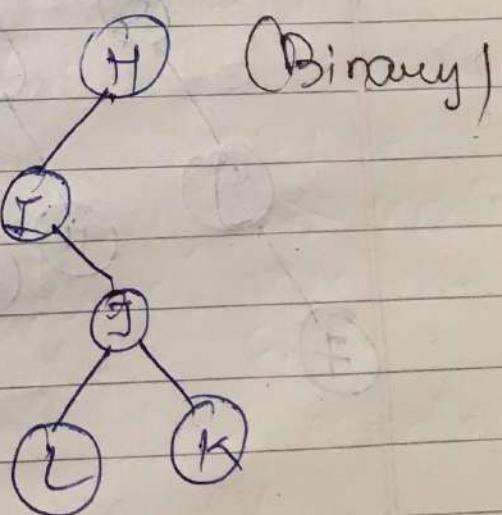
1) Root of general tree become a root of binary tree.

2) Find the 1<sup>st</sup> child node of a root node & attach it as a left child of parent node in a binary tree.

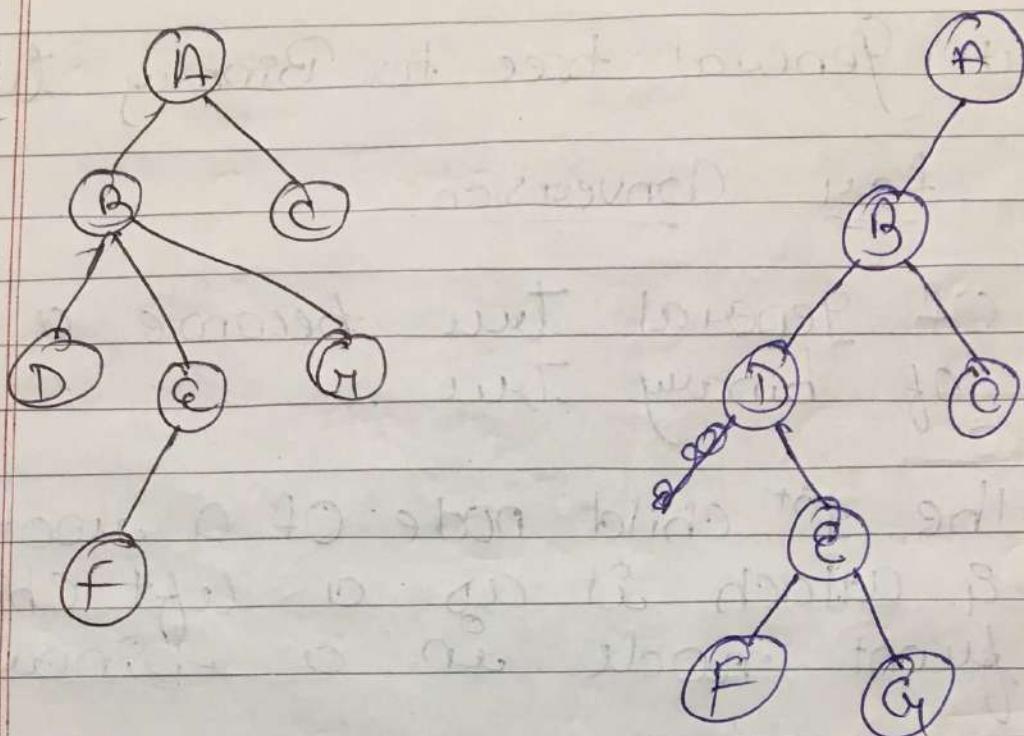
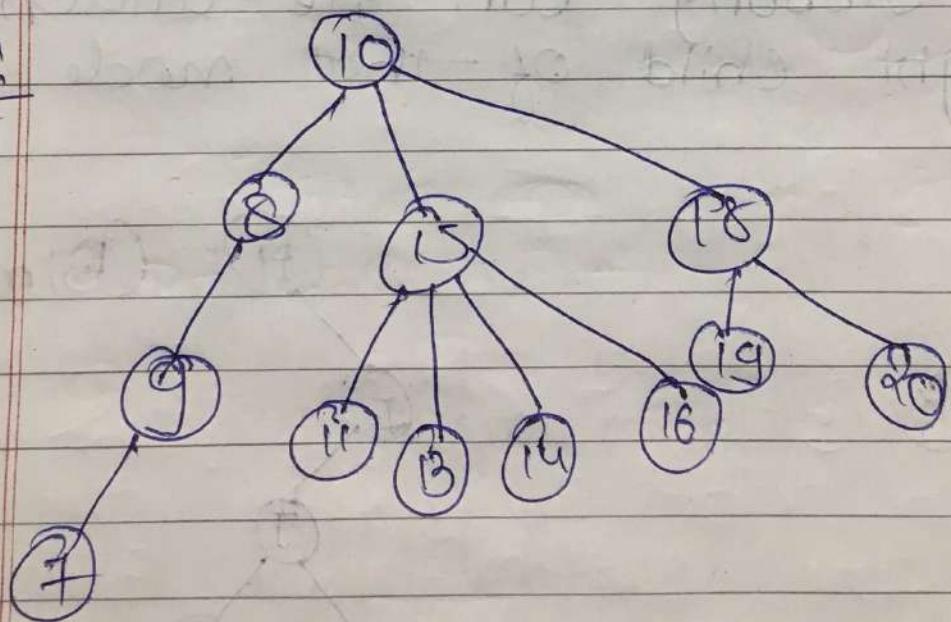
3) Right sibling can be attached as right child of that node.

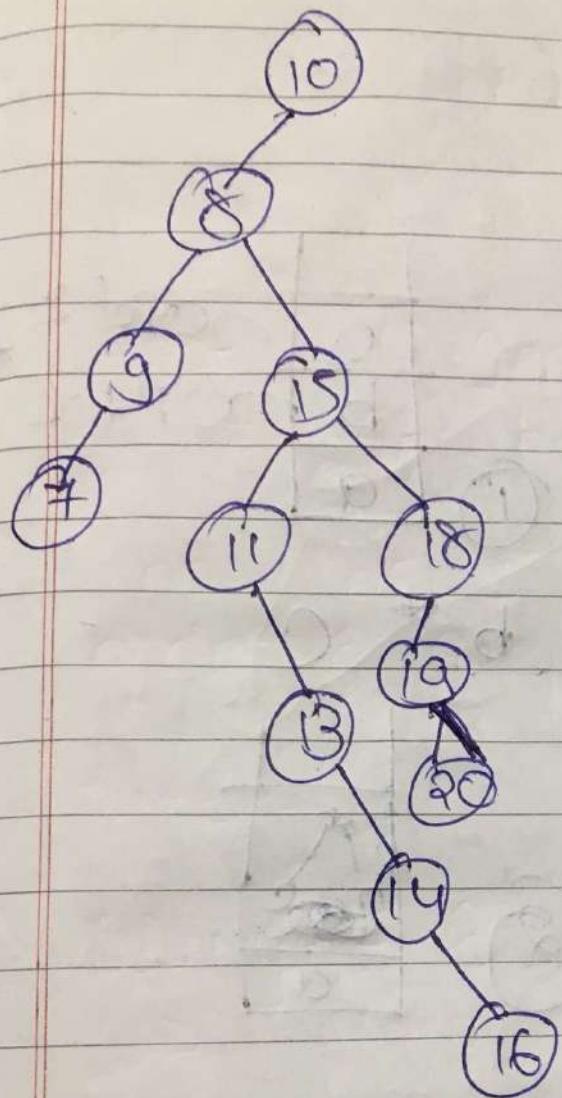


(General)



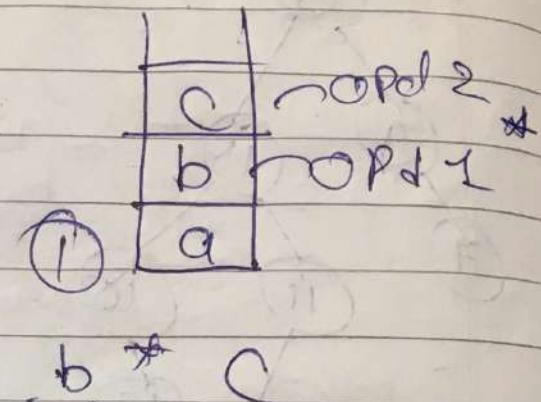
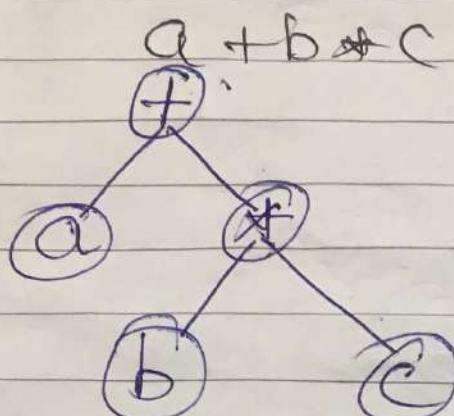
(Binary)

Ex:Ex:

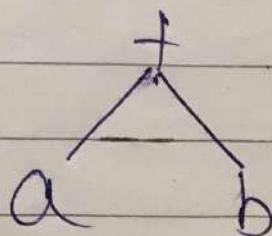
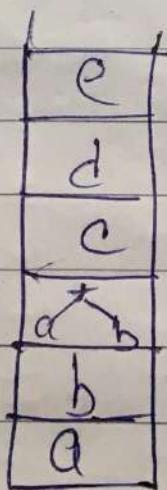


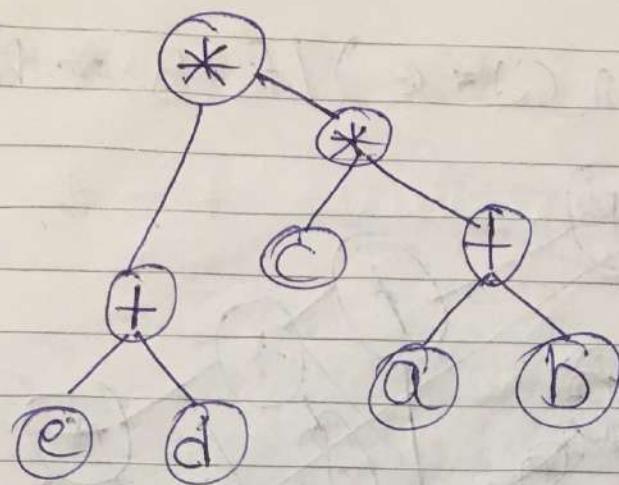
\* Expression tree

Ex:



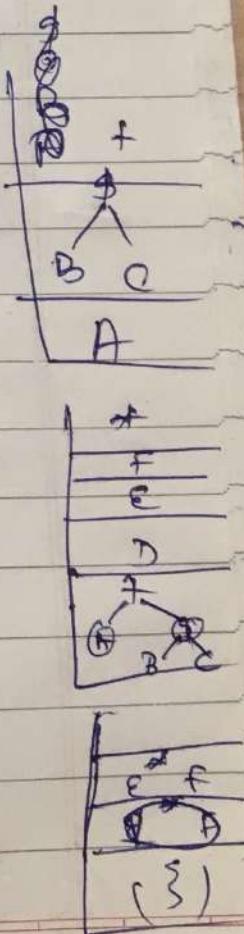
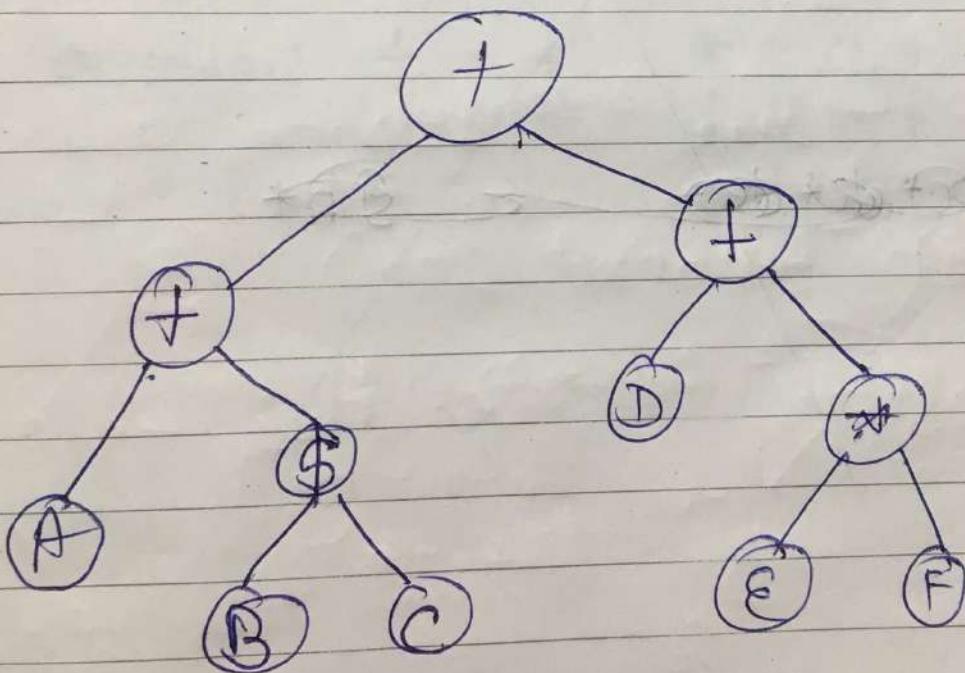
Ex: ab + cde + \* \*



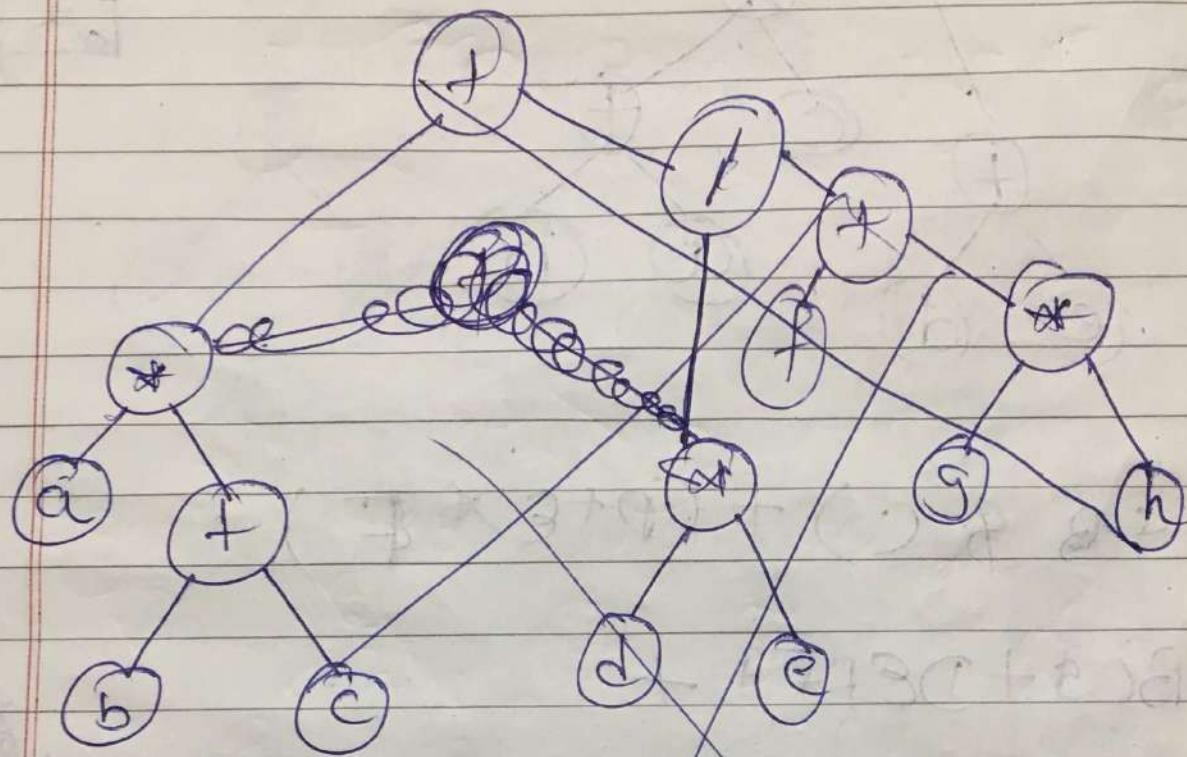


Ex:  $(A+B*C)+(D+E*F)$

= ABC\\$ + DEF\* + +

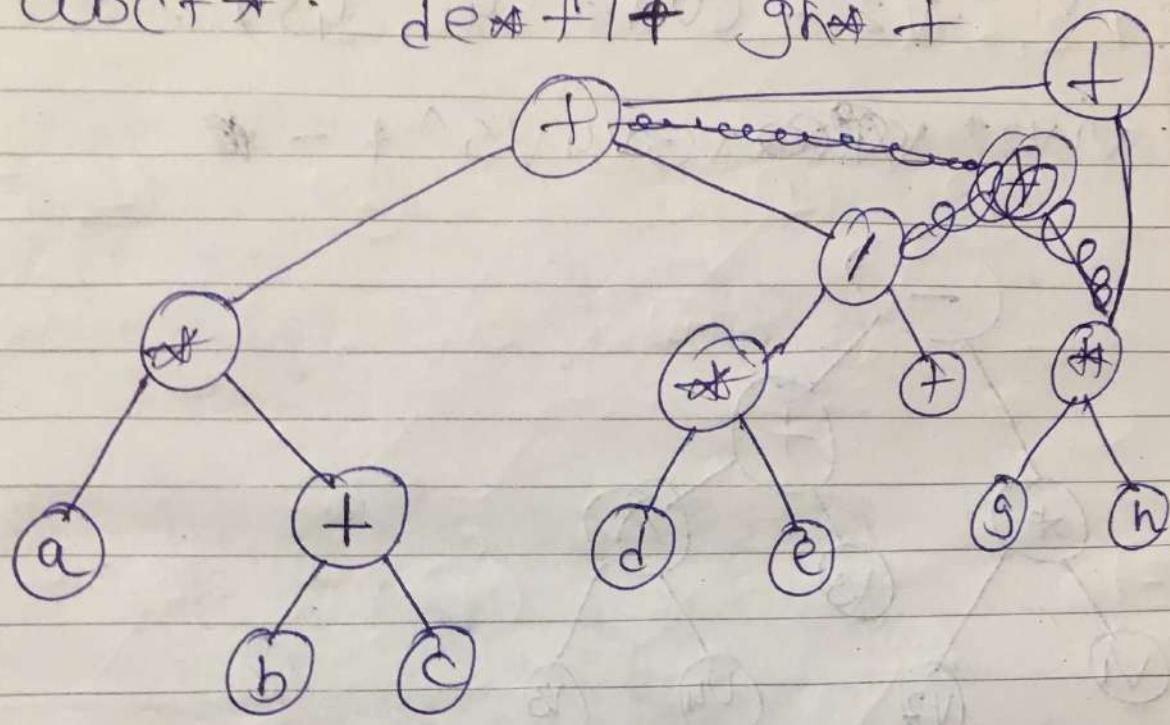


Ex:  $a * (b + c) + (d * e) / f + g \rightarrow h$



~~b c + d e \* f / g + h \*~~

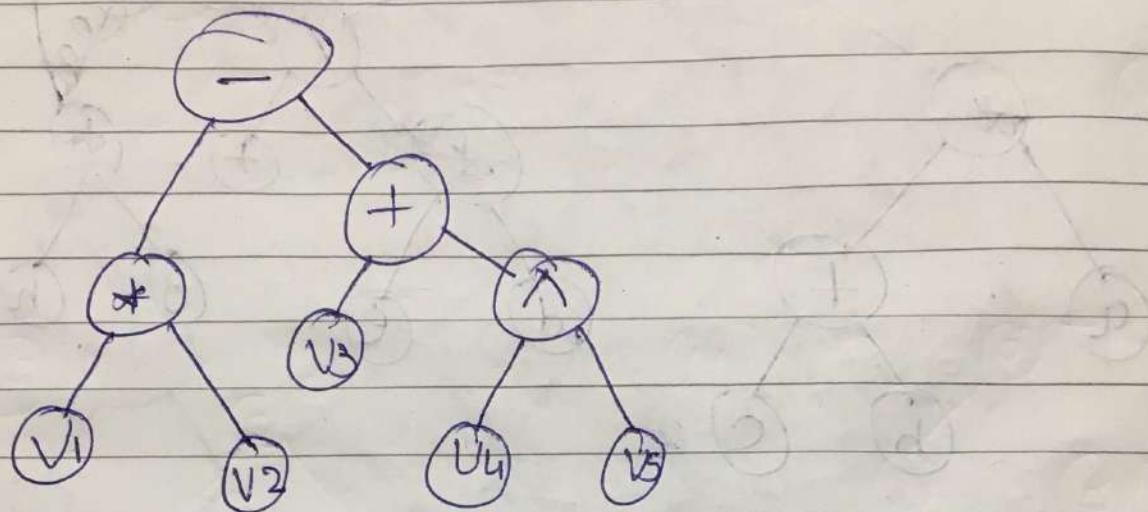
Ex: abc + \* . def + / + gh \* +



Permutation: + - + - \* - a - + - b - c -  
/ - \* - d - e - f - \* - g - h

Ex:  $V_1 * V_2 - (V_3 + V_4)^{1/5}$

$= V_1 V_2 * \cancel{V_3} \cancel{V_4} V_5^{1/5} + - \cancel{\#}$



\* AVL Tree - (Adelson, Velski, Landis)  
(Height Balanced Tree)

BF (Balance Factor) =  $H_L - H_R$   
(height of left Sub-tree) (height of right Sub-tree)

AVL tree  $\rightarrow$  BF = 0, 1, -1

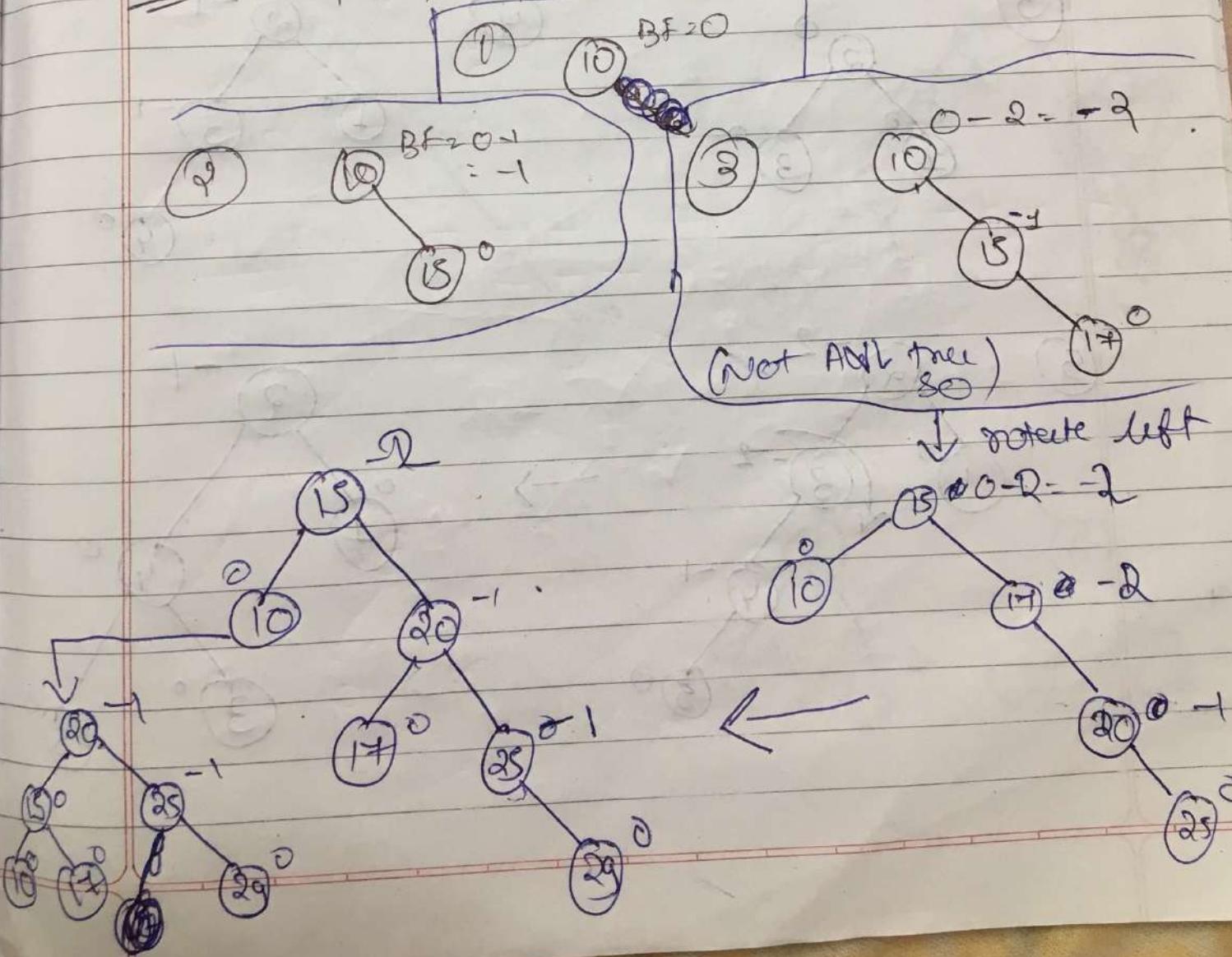
Date \_\_\_\_\_  
Page \_\_\_\_\_

height of right hi  
insert  
rotate  
problem

## Rotation

- ① left rotation }
  - ② right " }
  - ③ left-right " }
  - ④ right-left " }
- } Single rotation
- } Double "

Ex: 10, 15, 17, 20, 25, 29

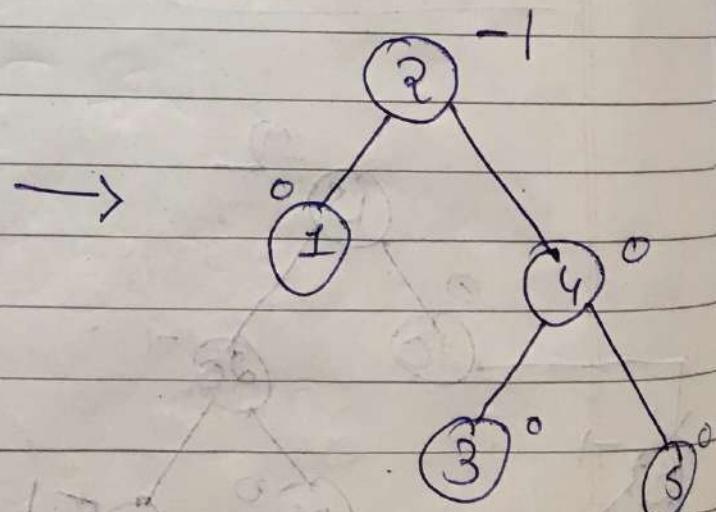
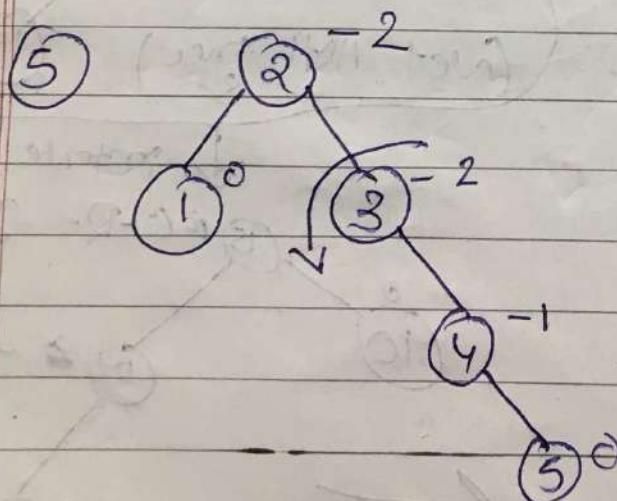
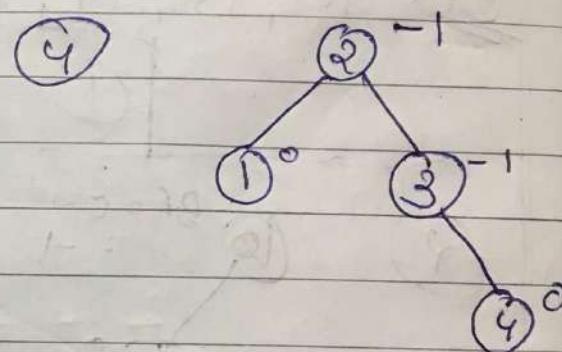
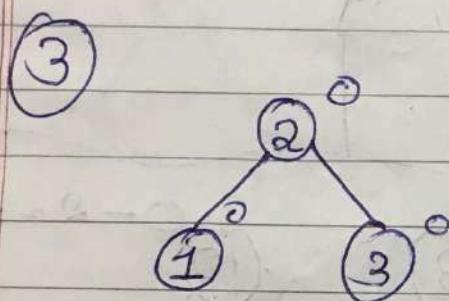
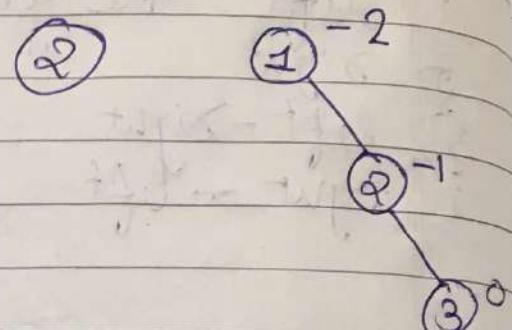
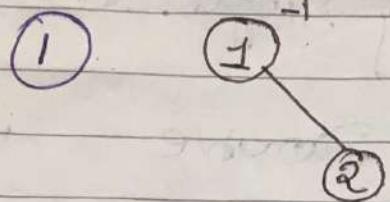


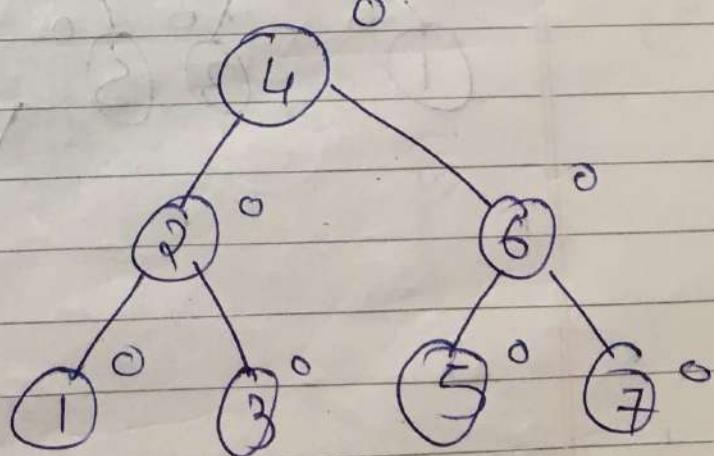
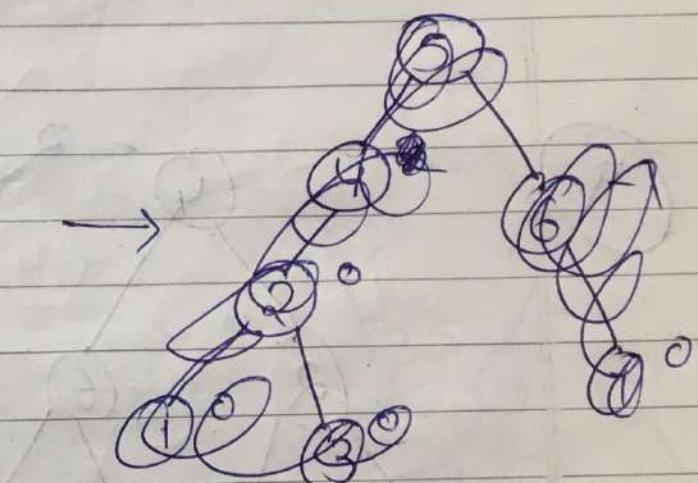
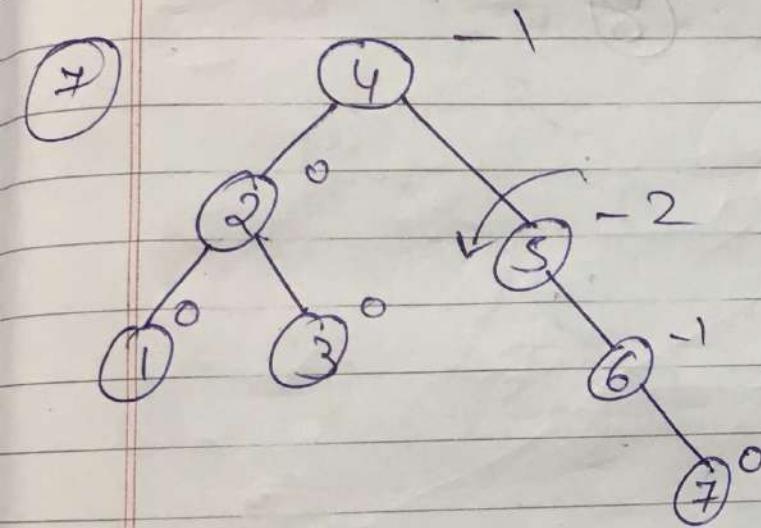
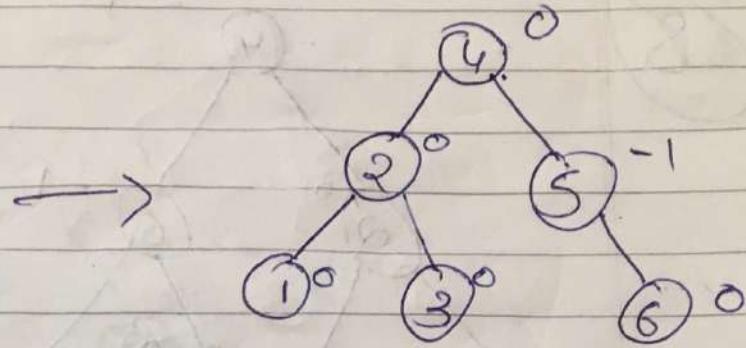
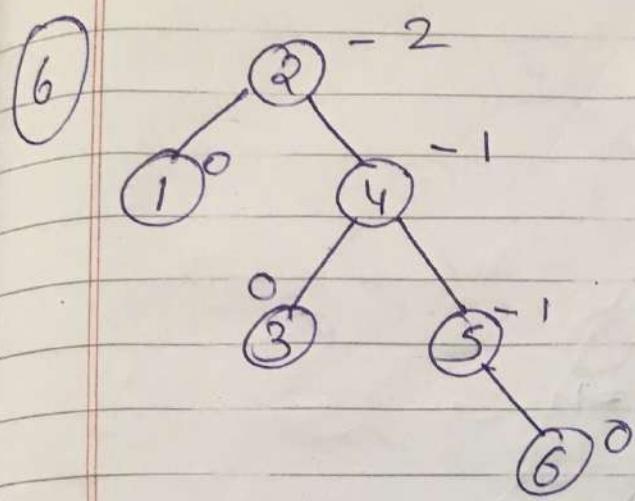
24/9/19

Date \_\_\_\_\_  
Page \_\_\_\_\_

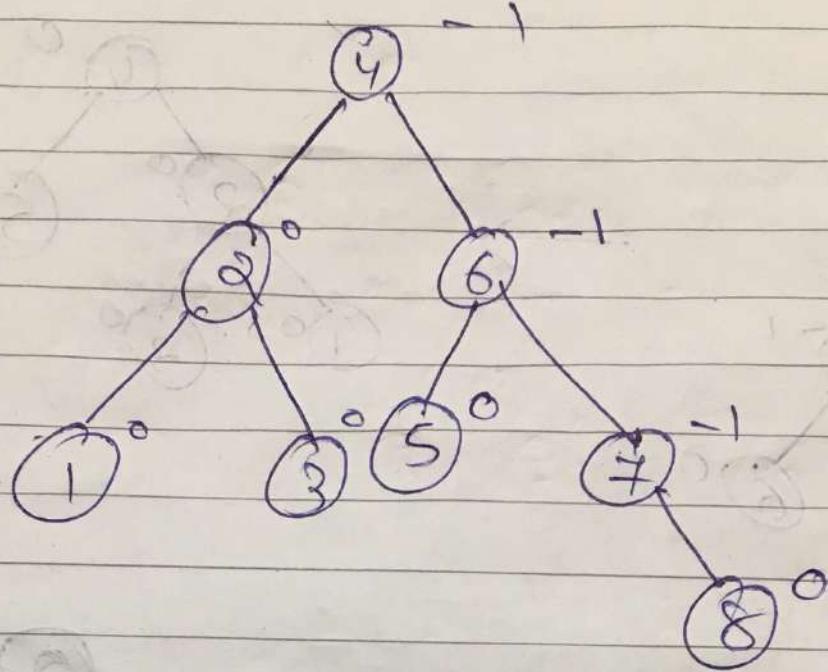
Ex:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

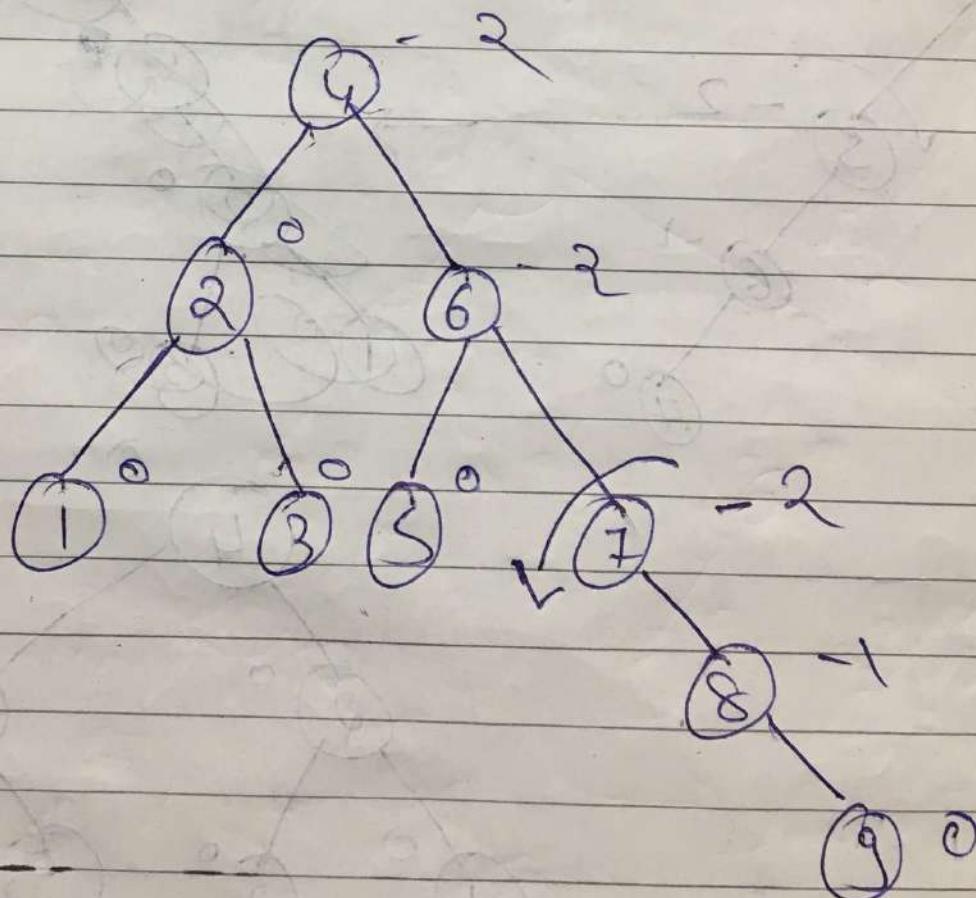


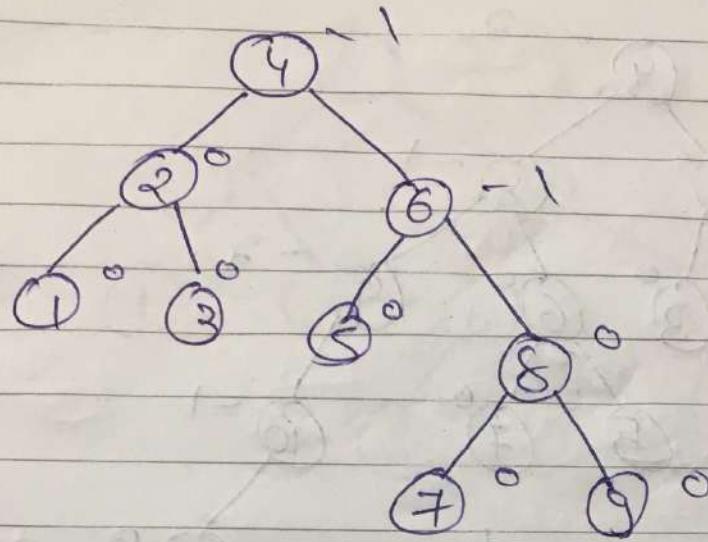


⑧

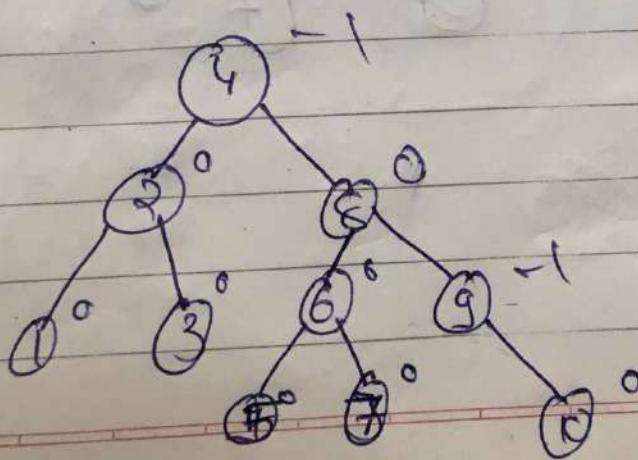
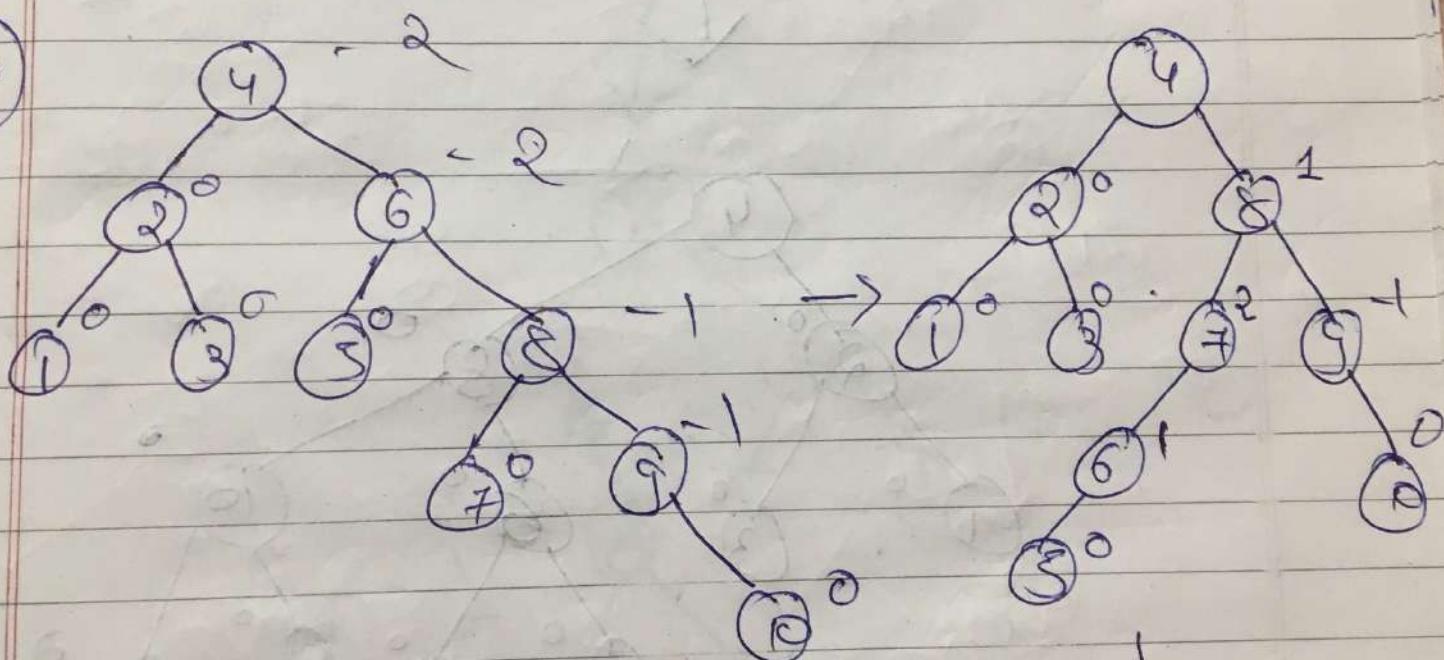


⑨

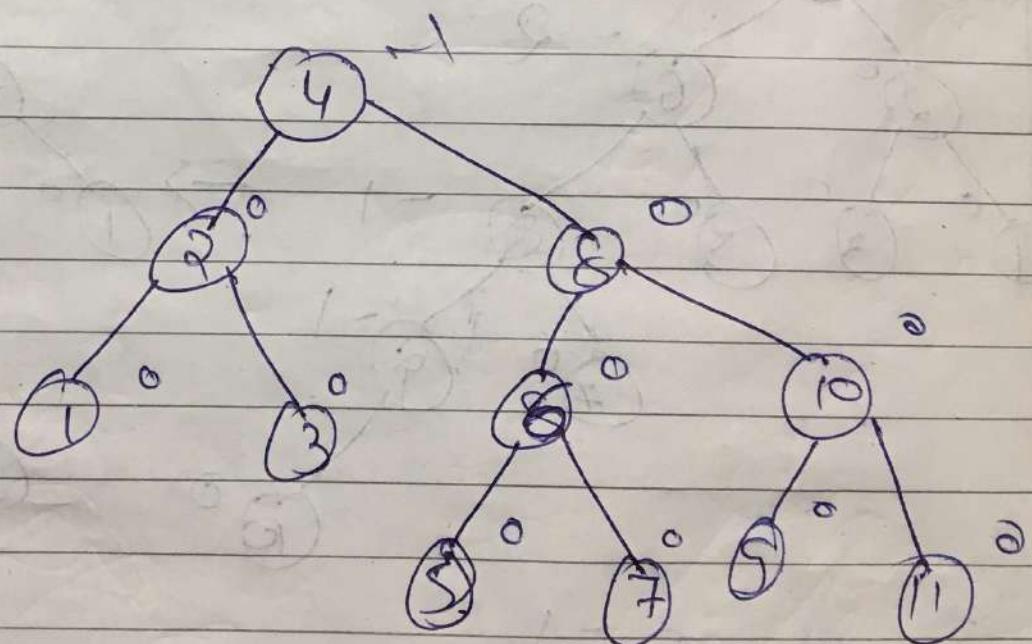
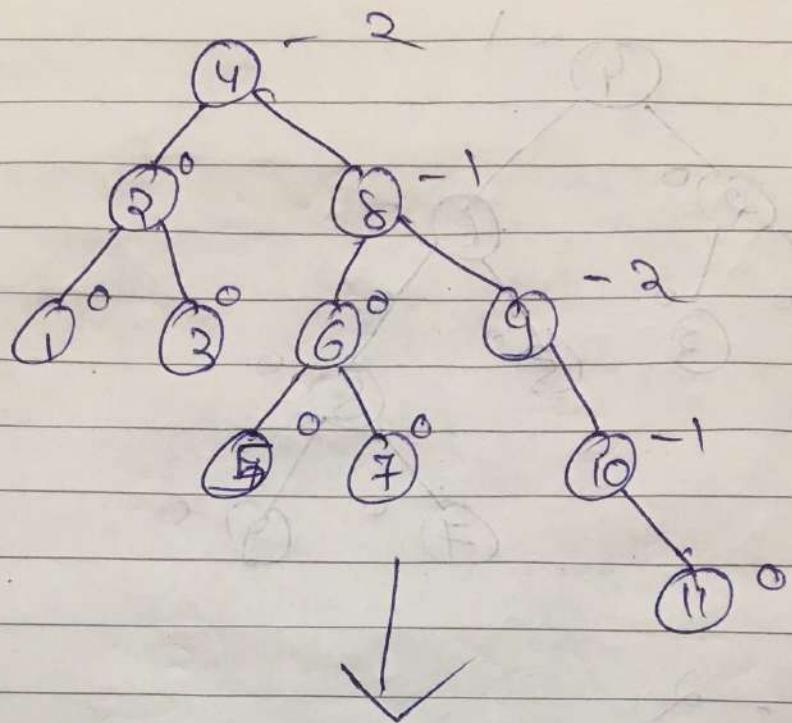




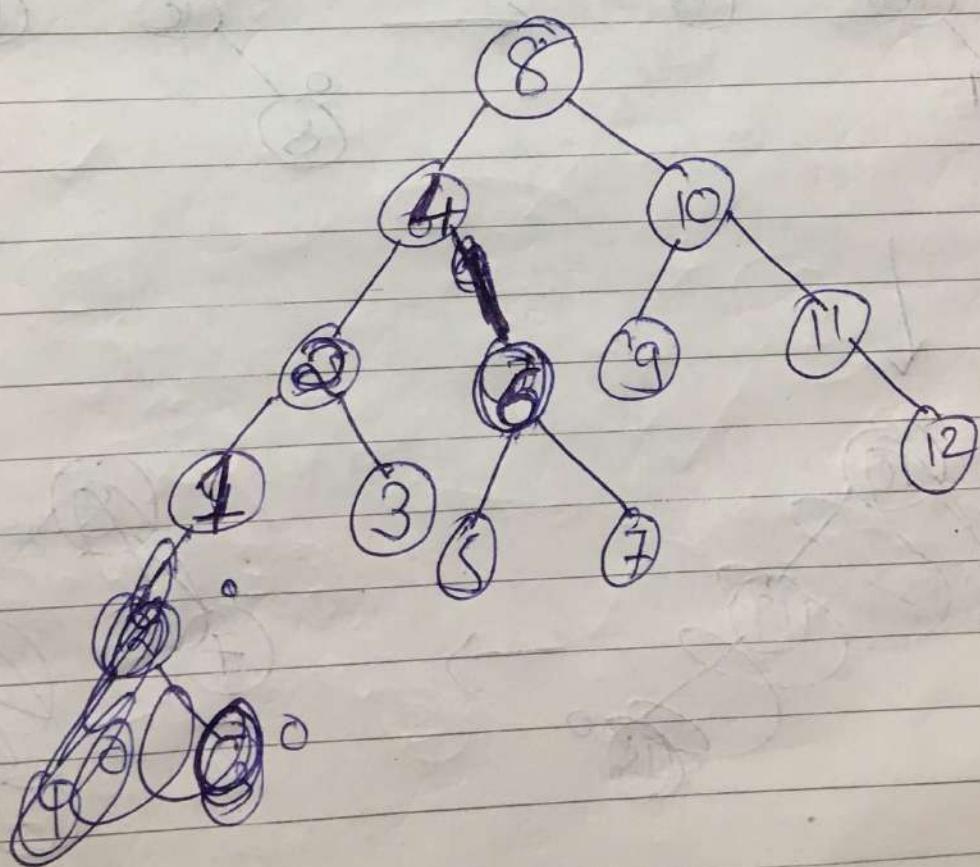
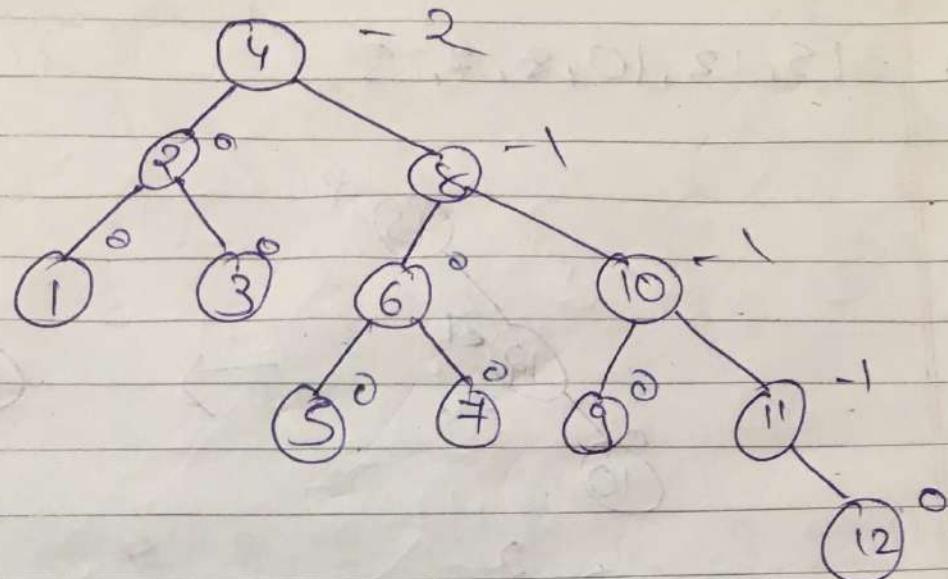
10



11

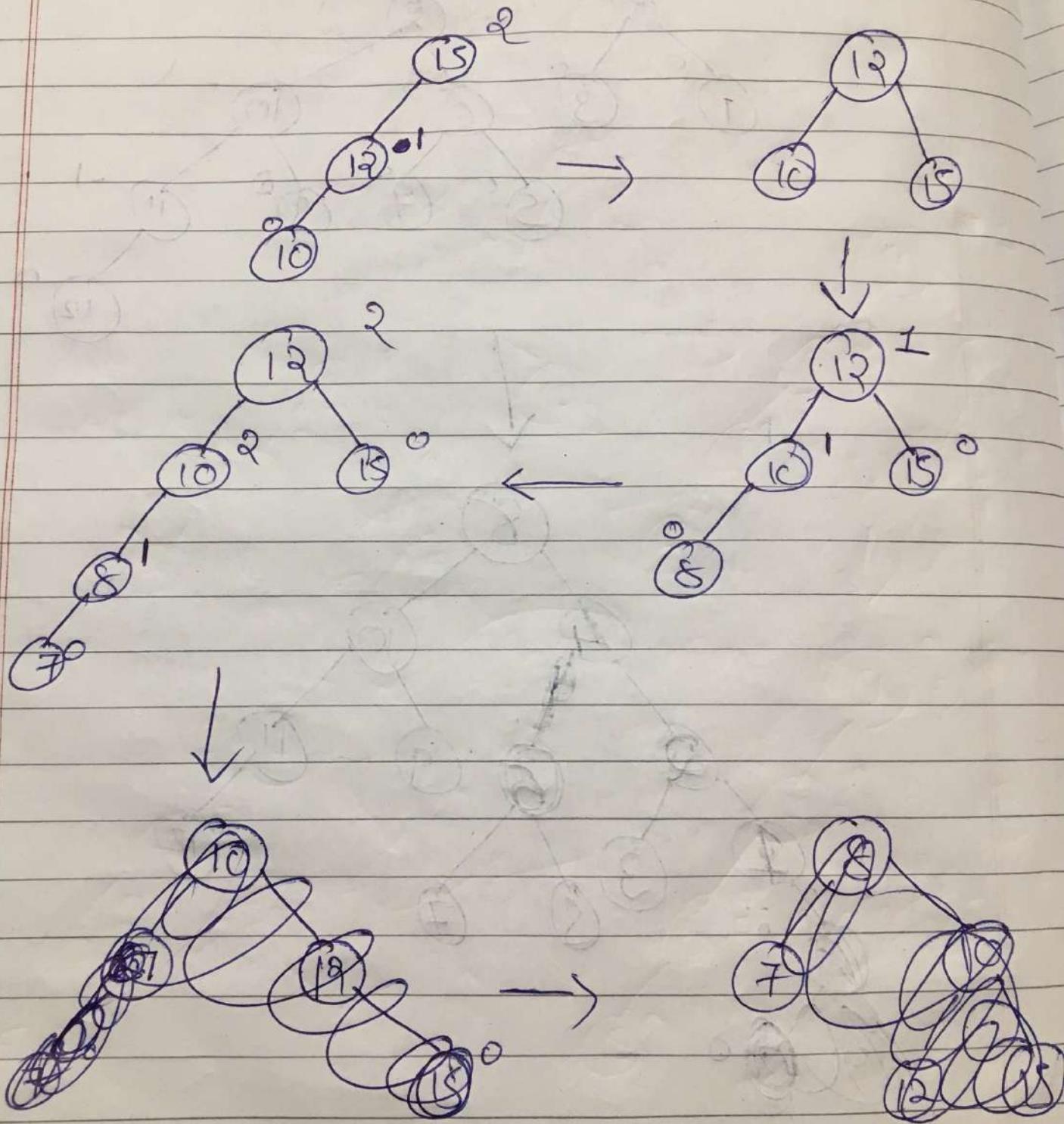


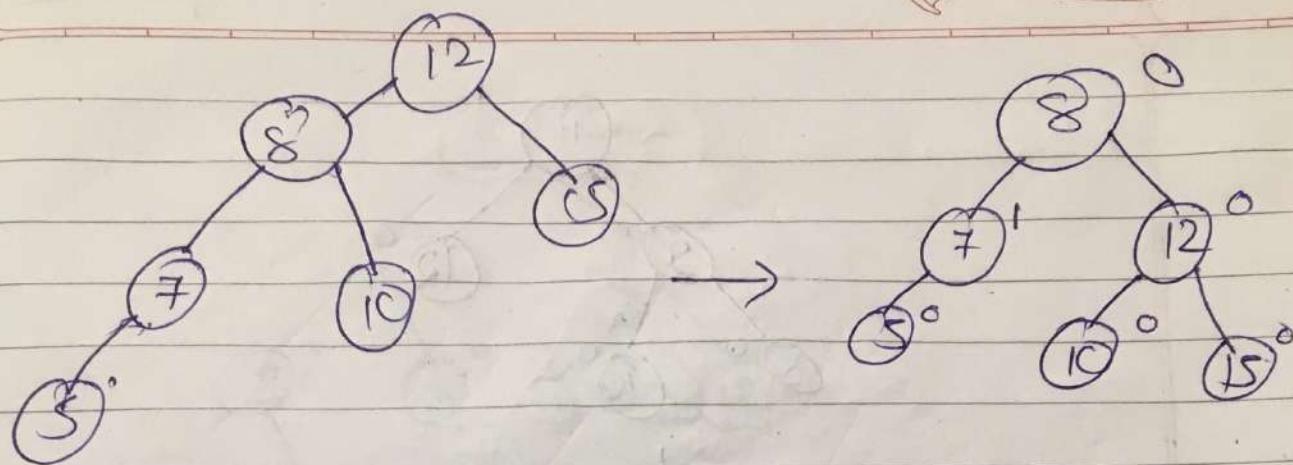
12



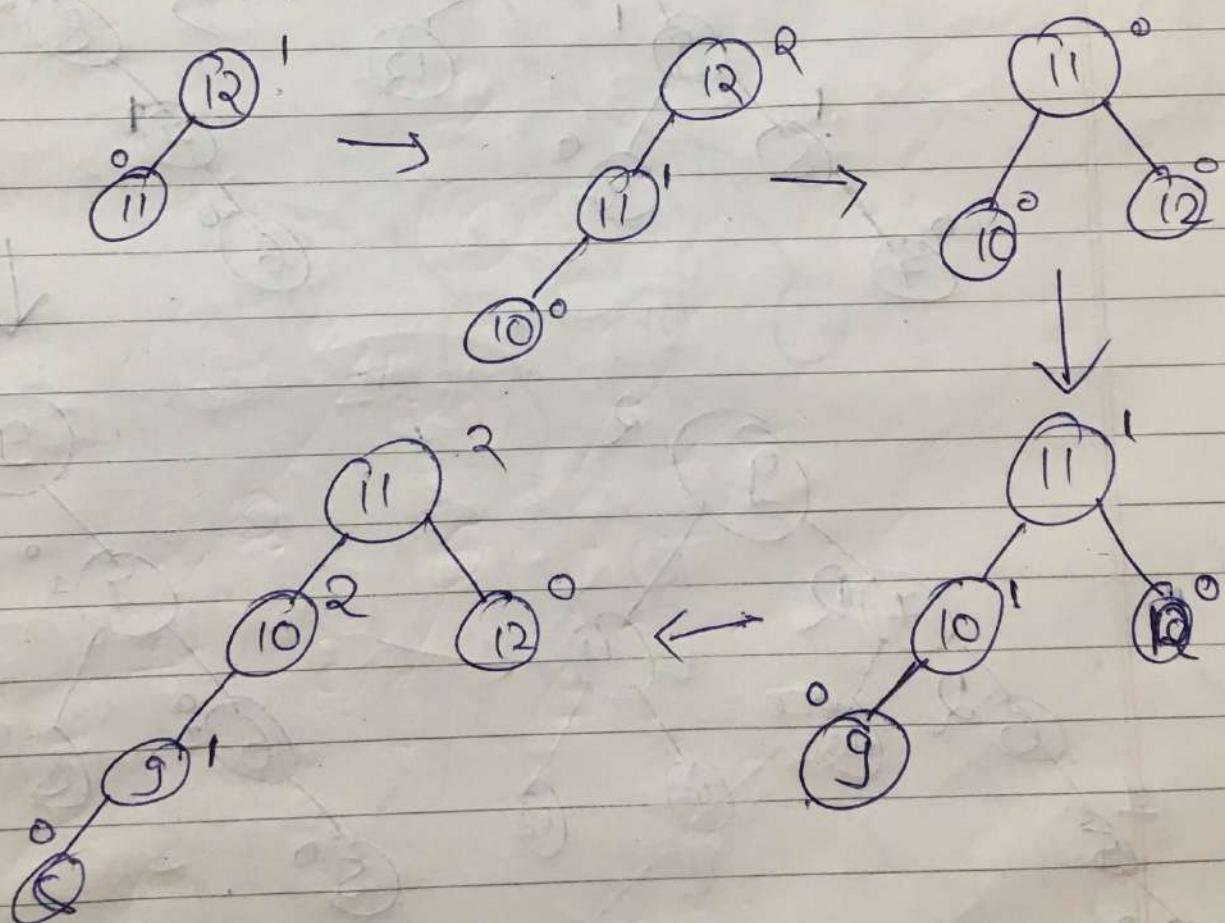
Ex:

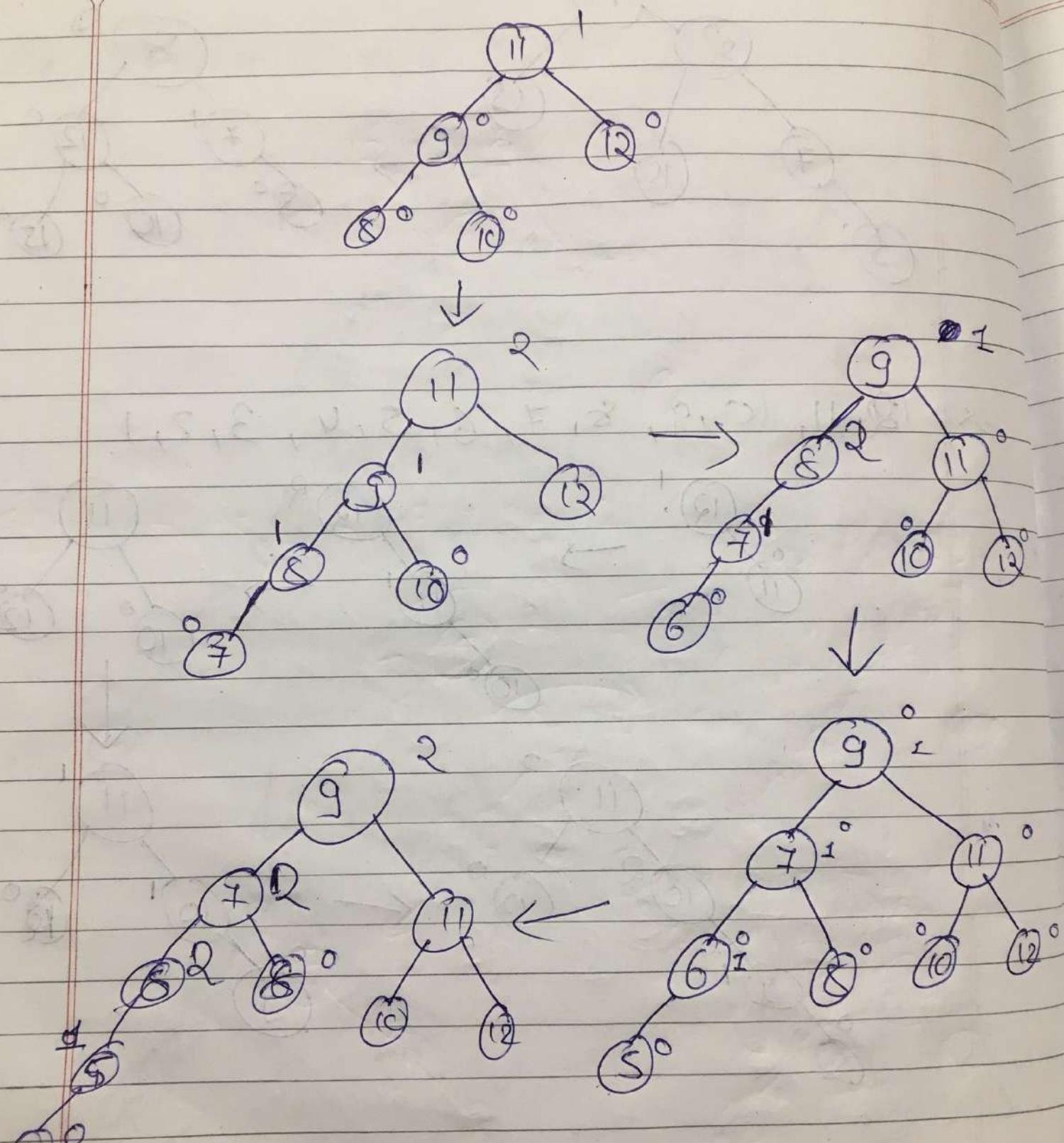
15, 12, 10, 8, 7, 5

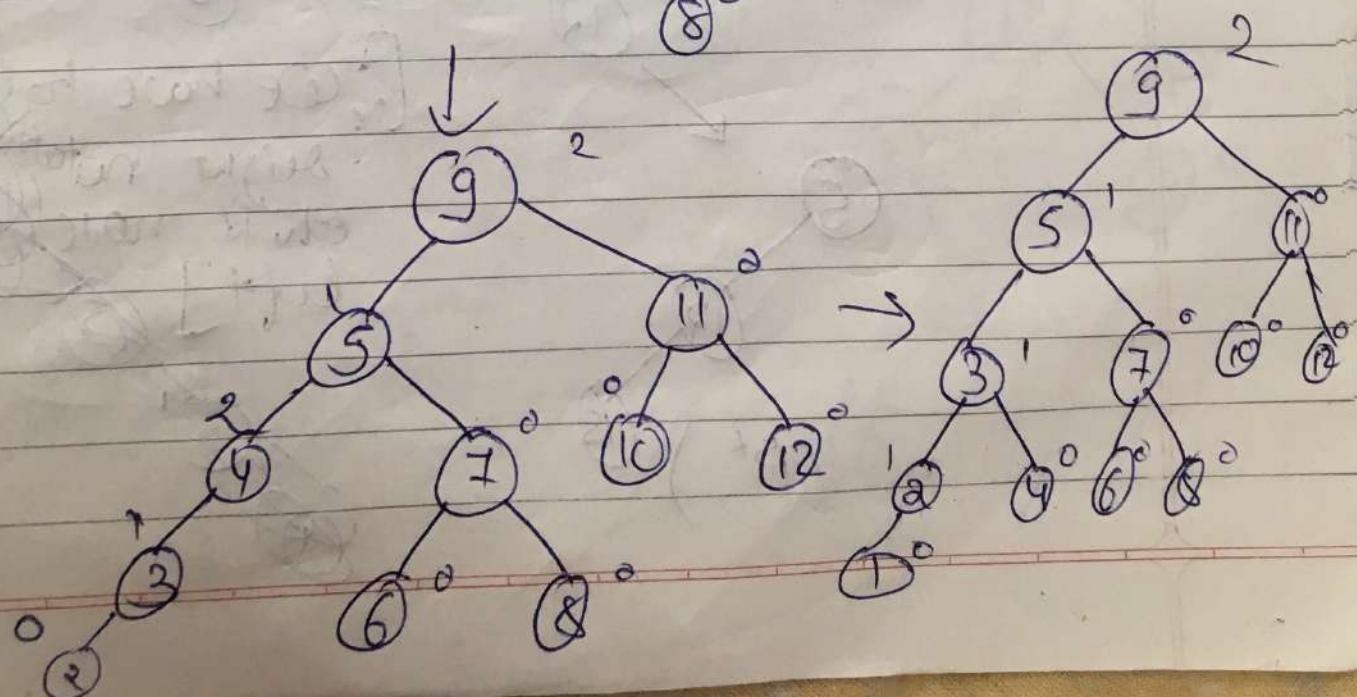
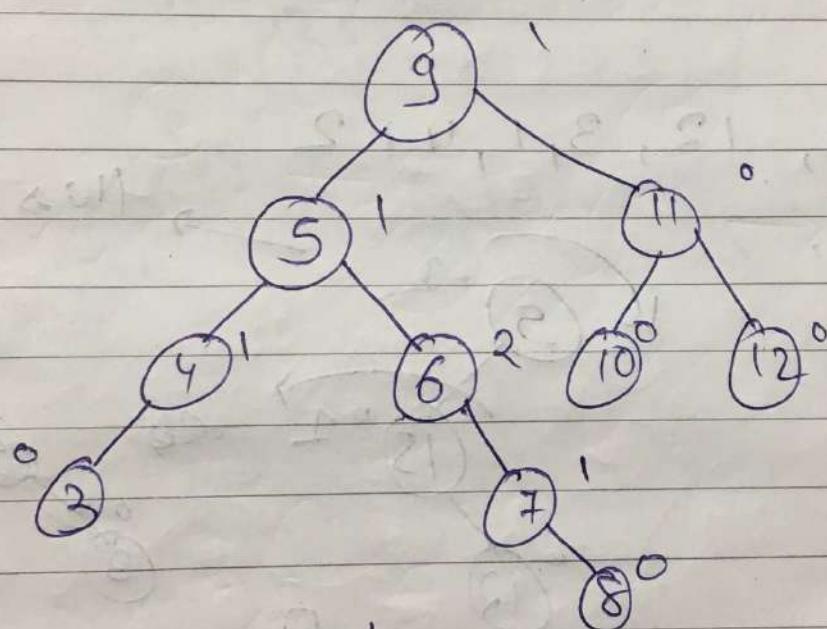
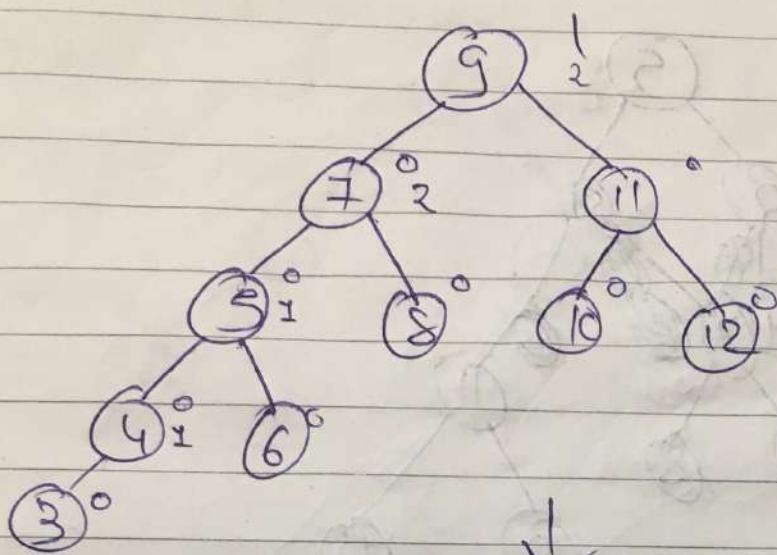


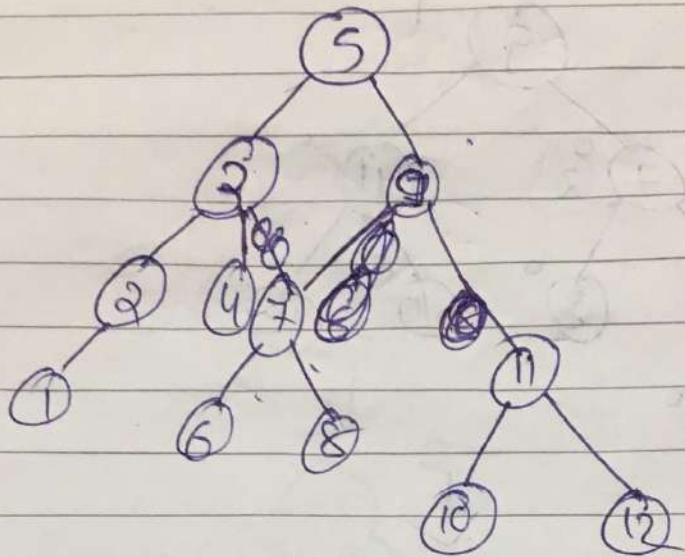


Ex 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1



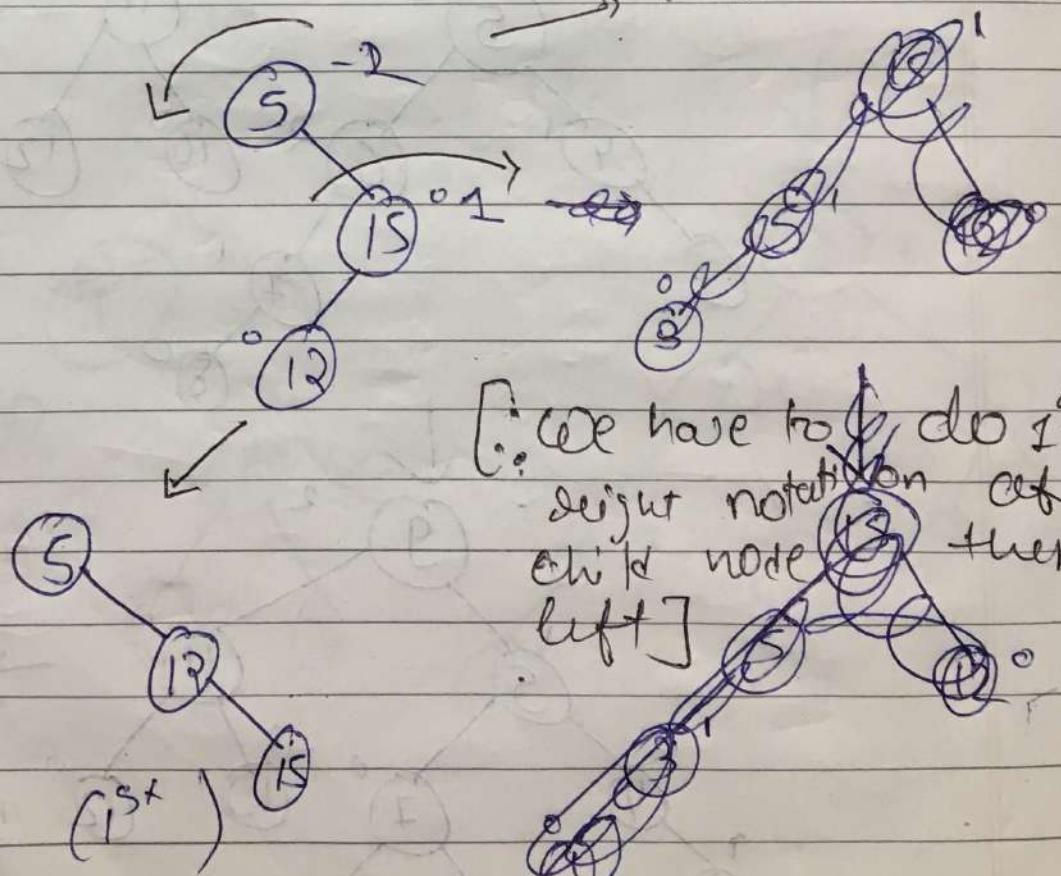




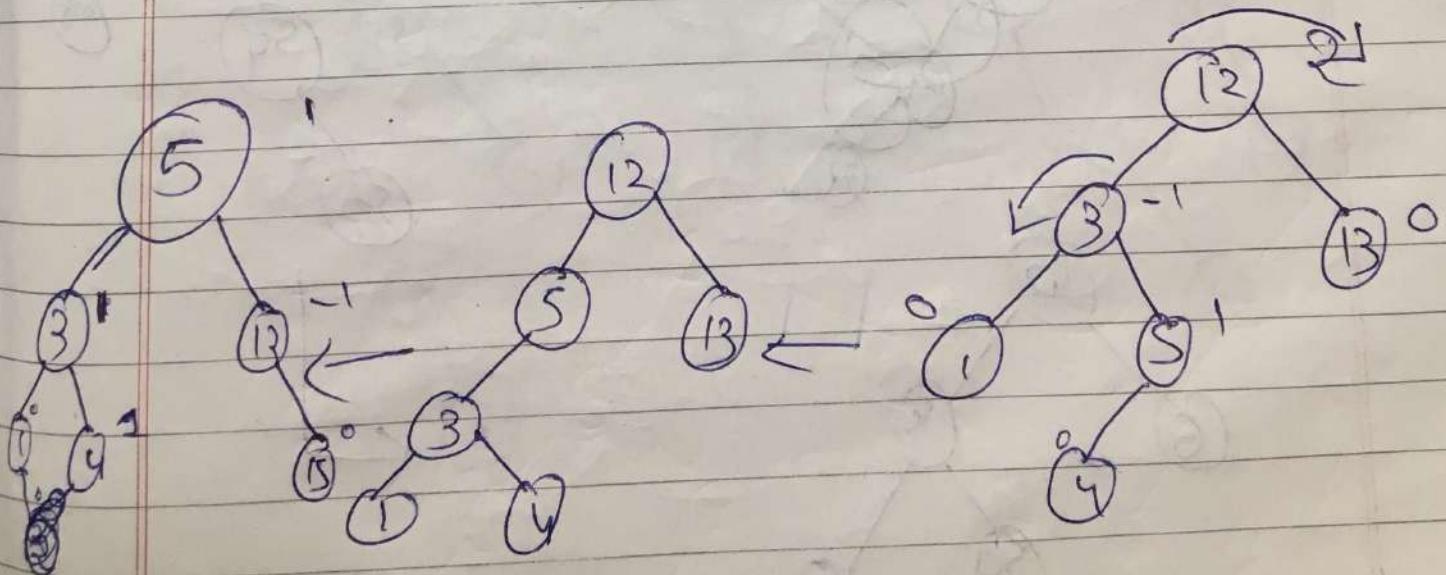
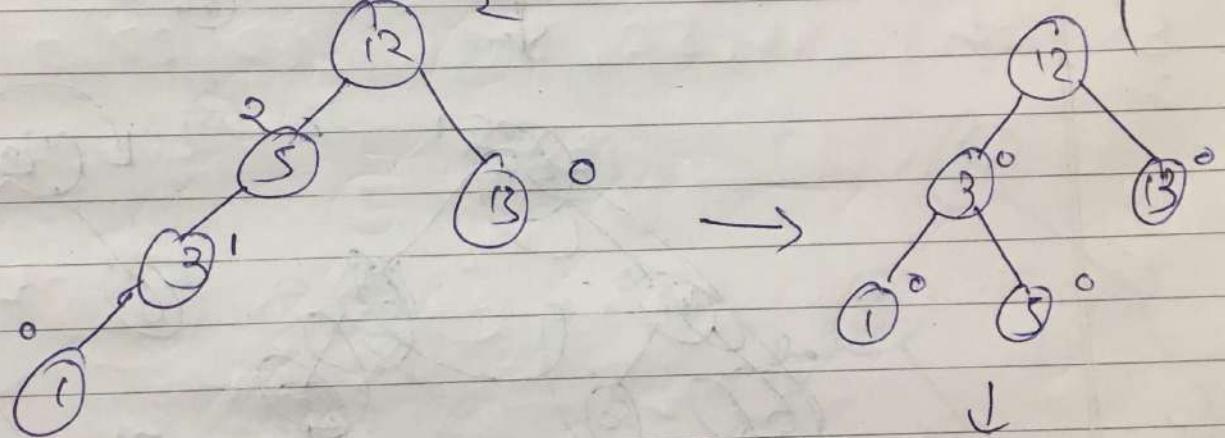
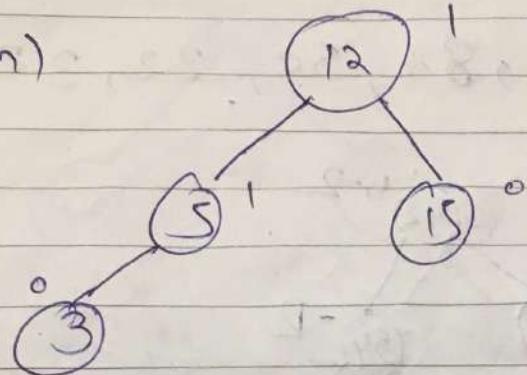


Ex: 5, 15, 12, 3, 1, 4, 2

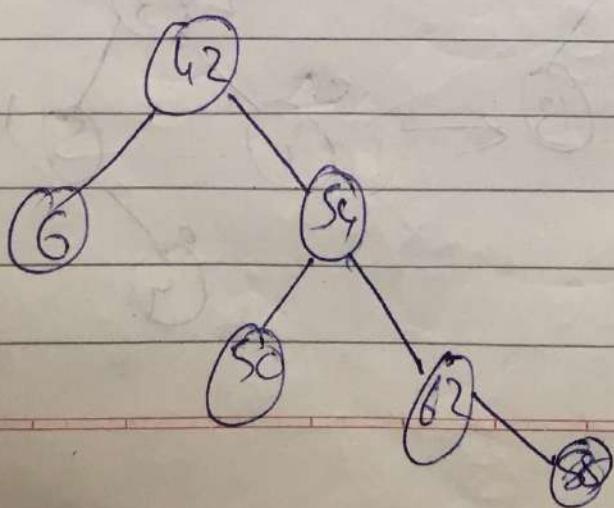
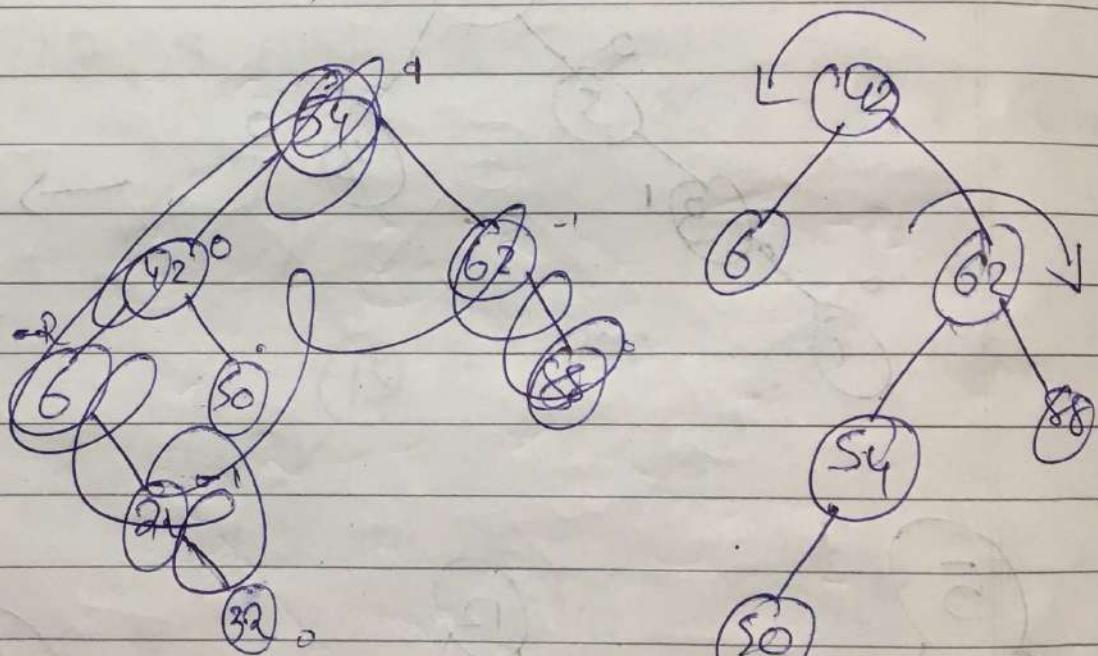
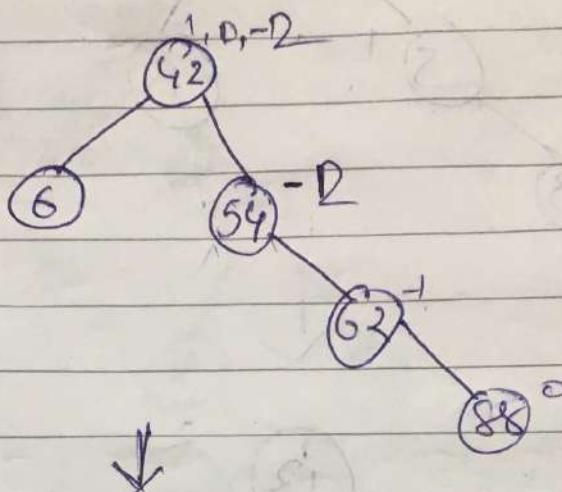
This is RL Rotation

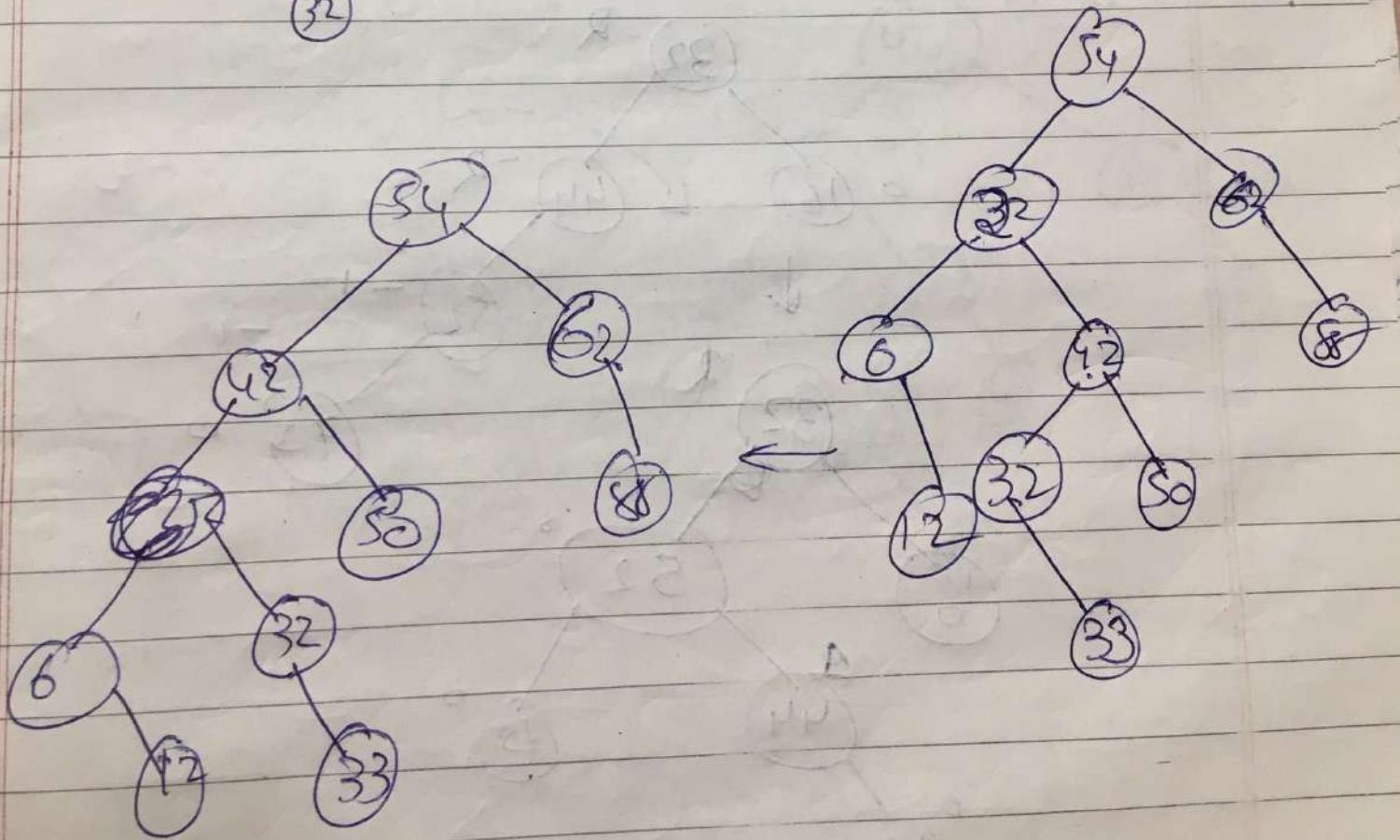
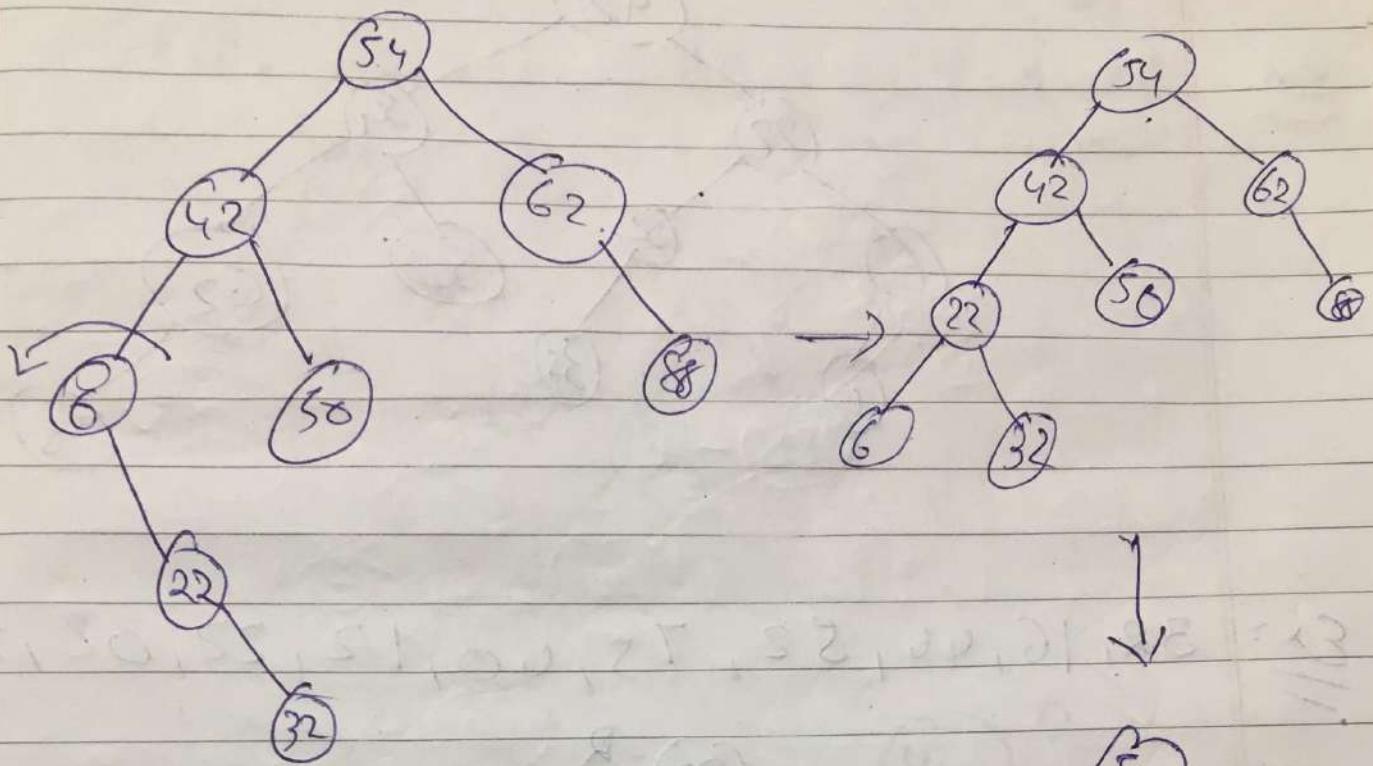


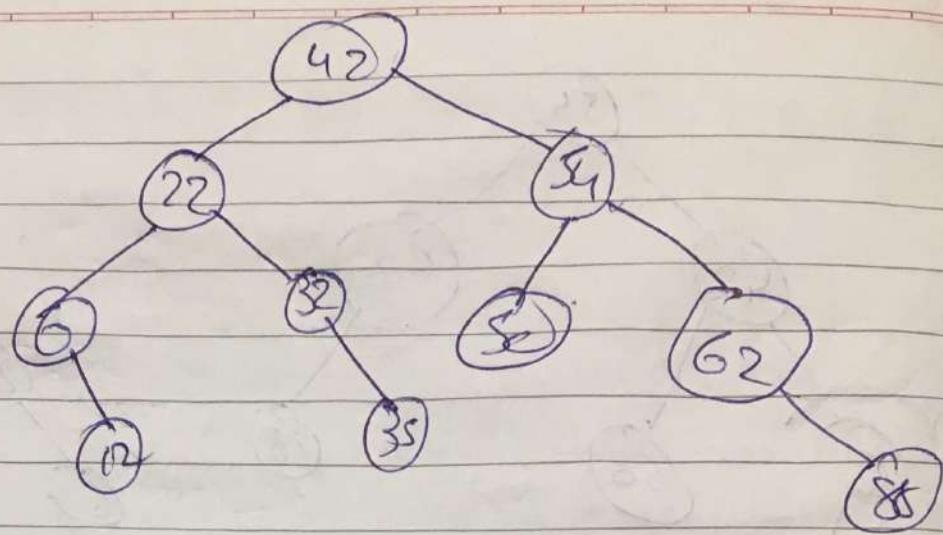
(then)



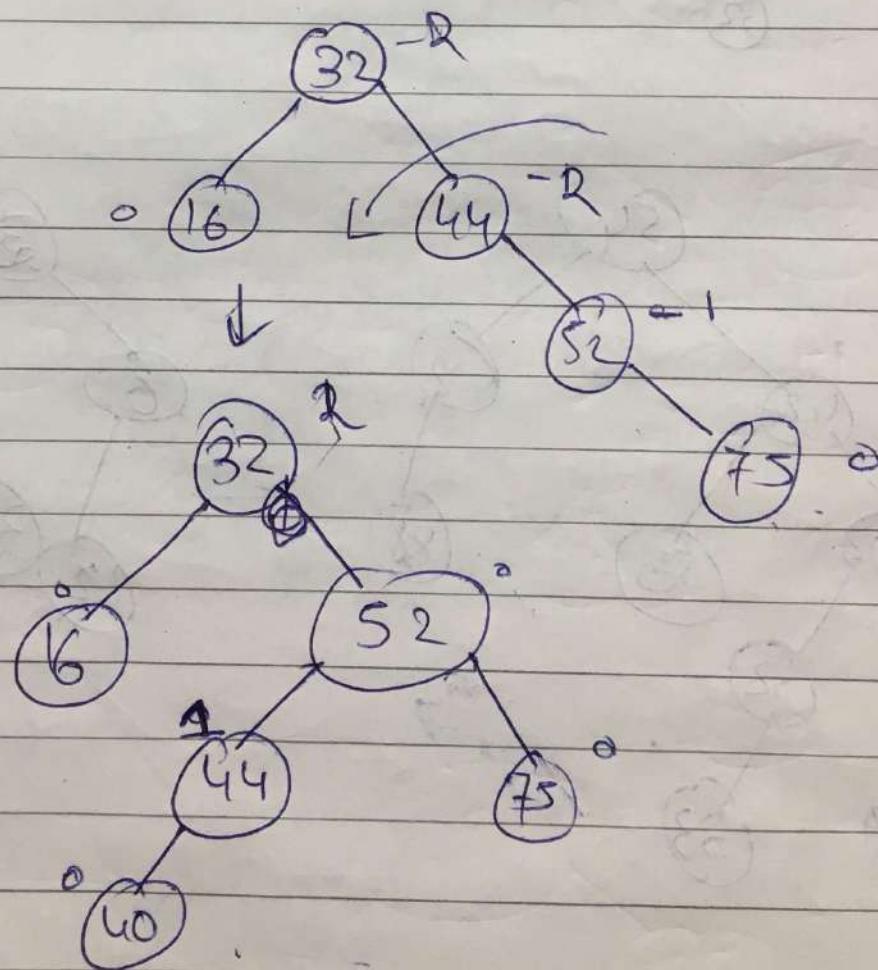
Ex 42, 6, 54, 62, 88, 50, 22, 32, 12, 33

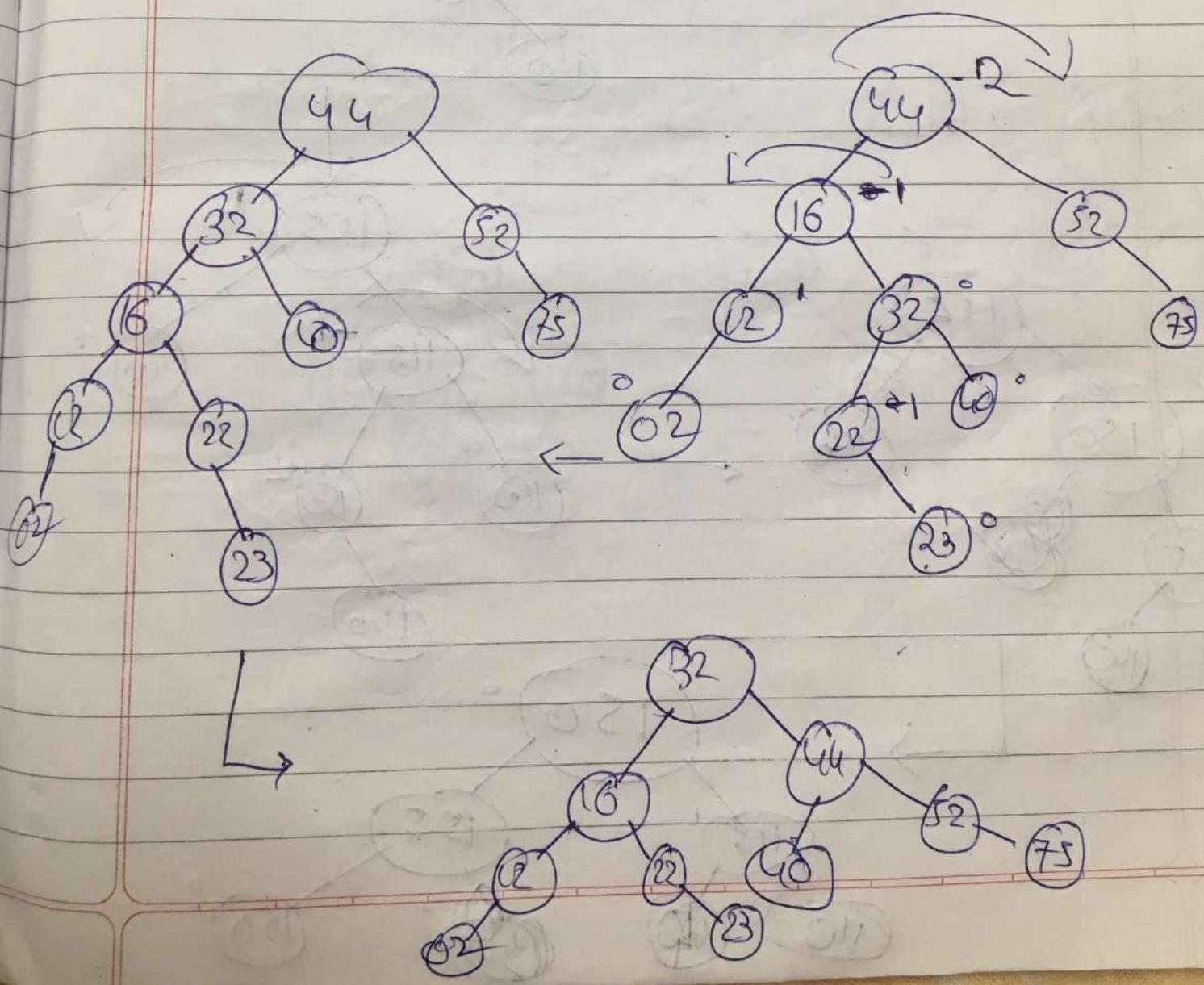
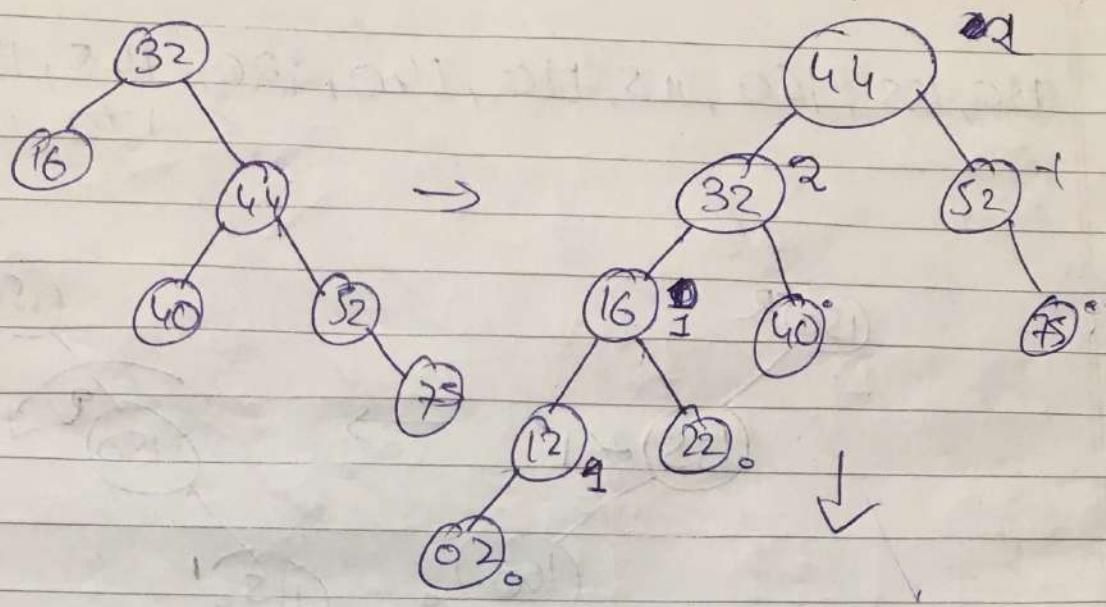




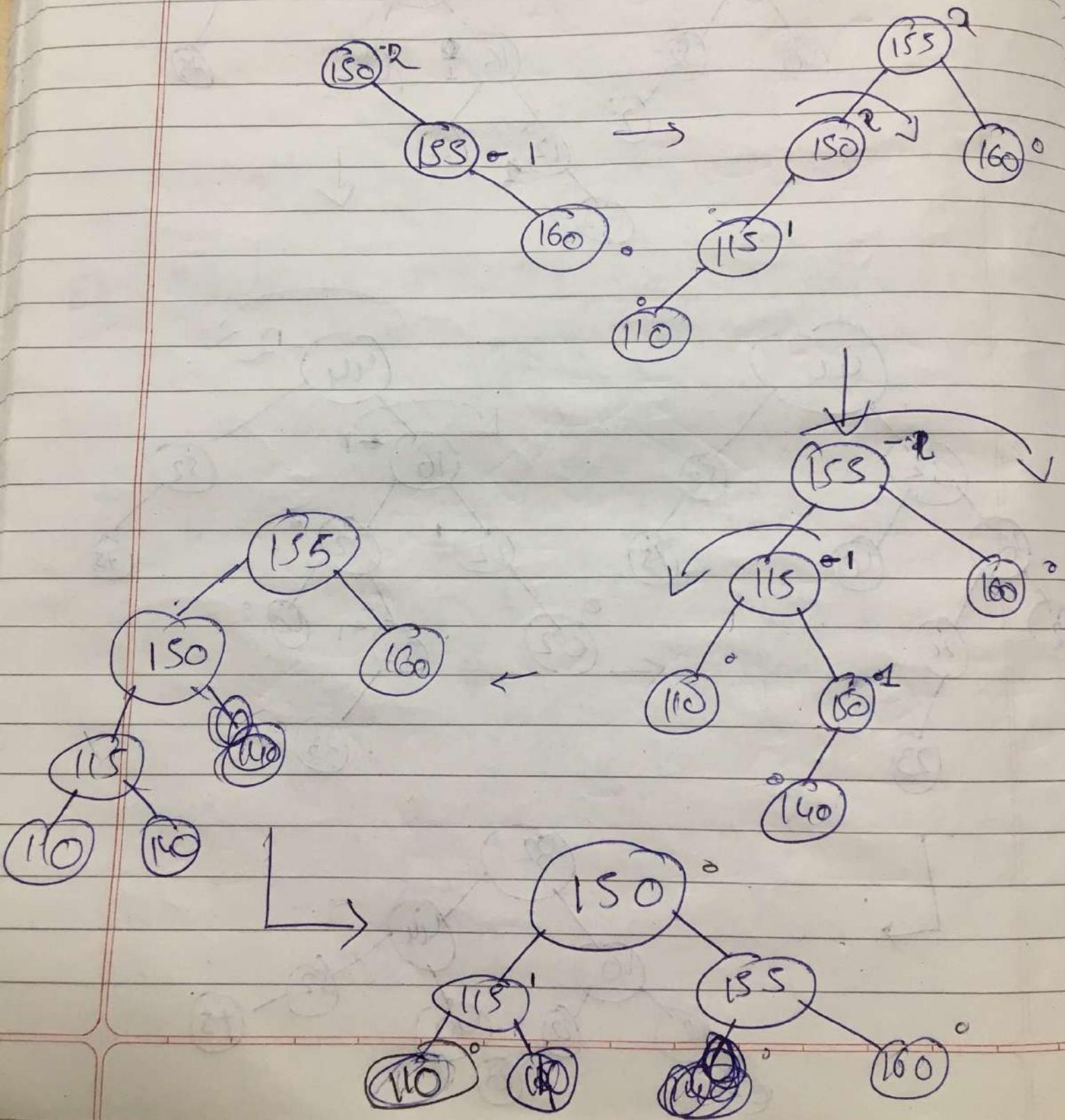


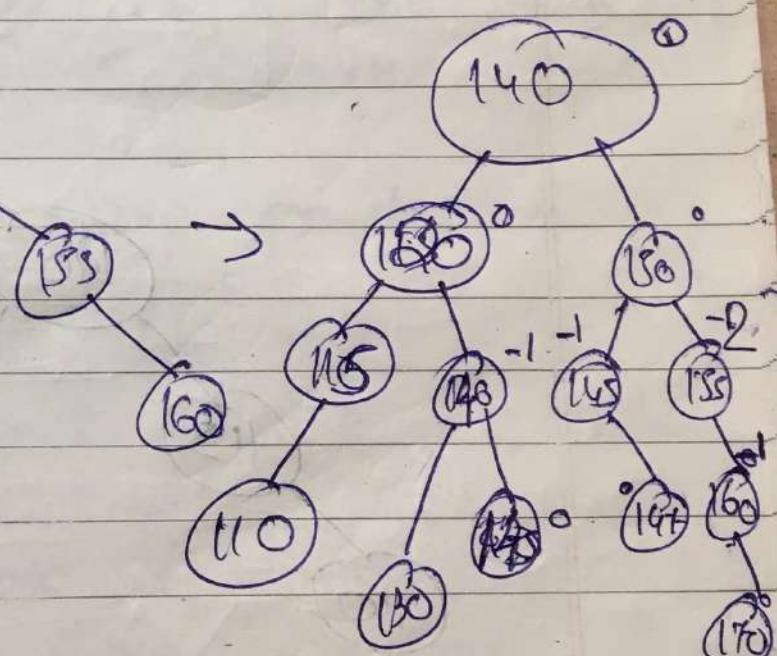
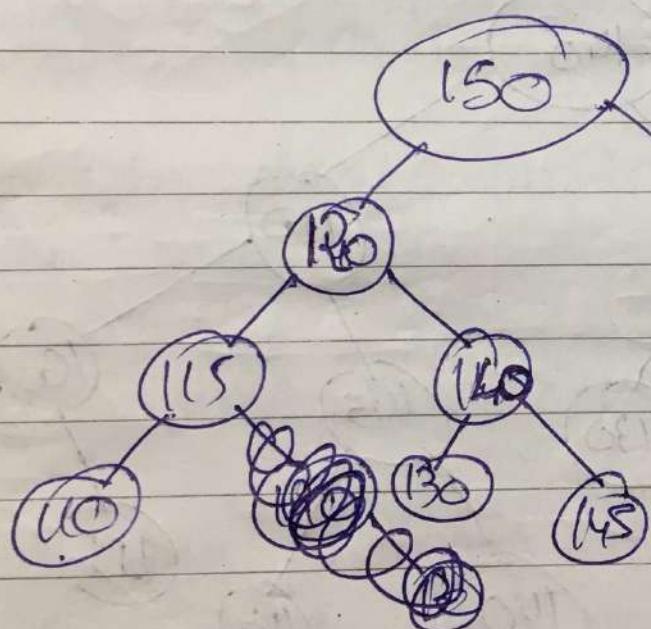
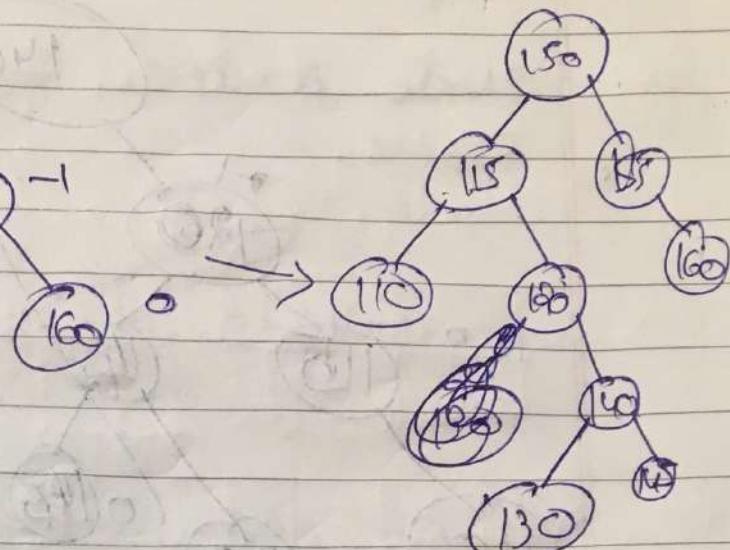
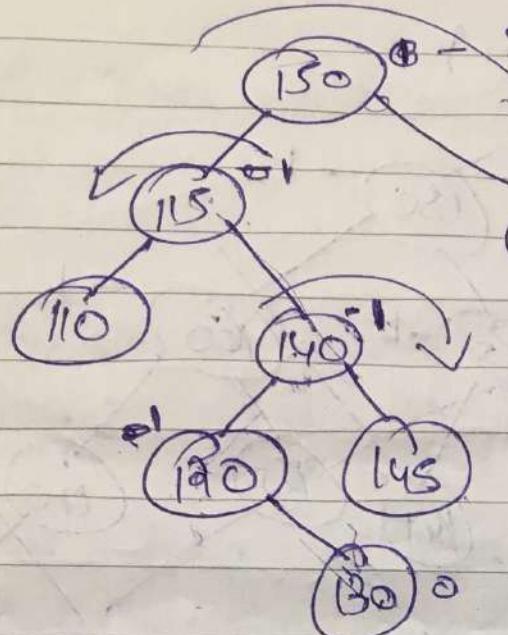
Ex: 32, 16, 44, 52, 75, 40, 12, 22, 02, 23

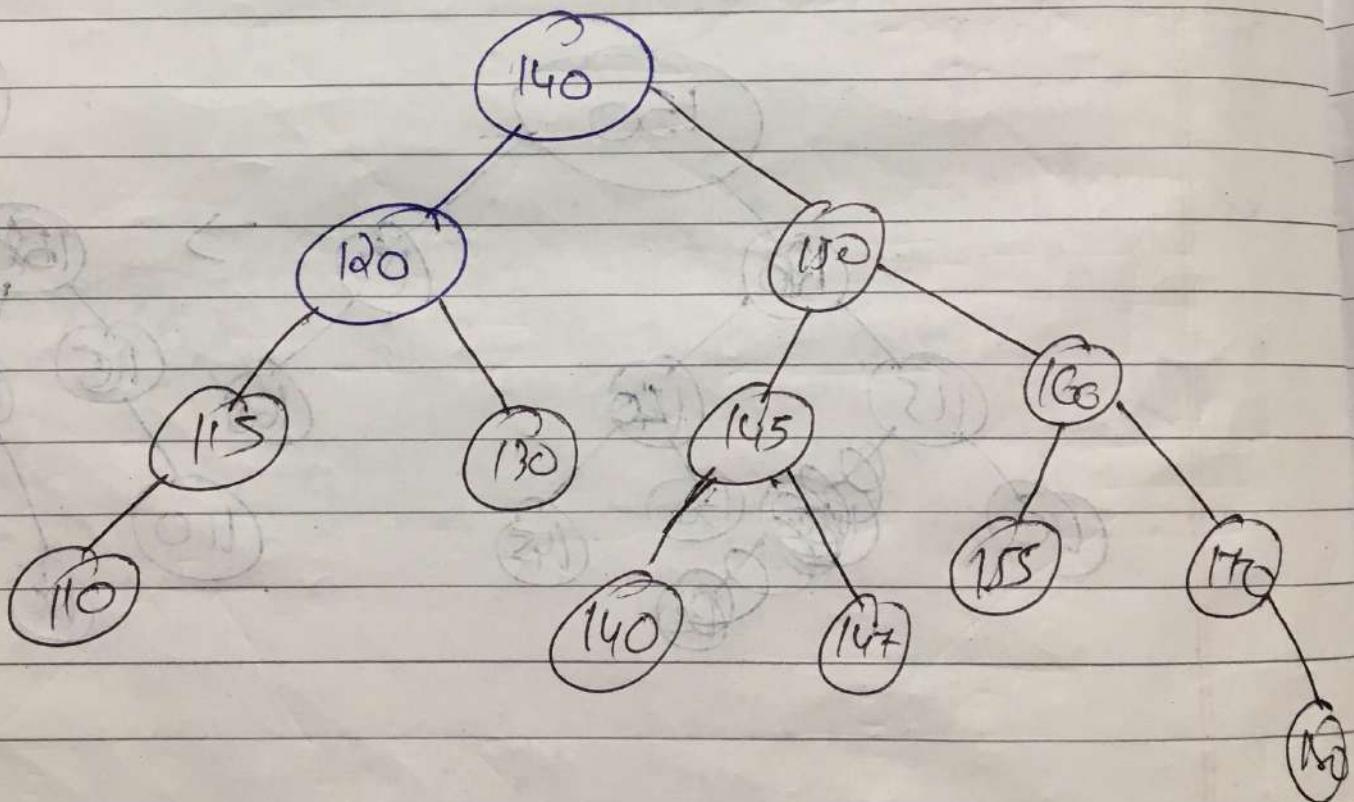
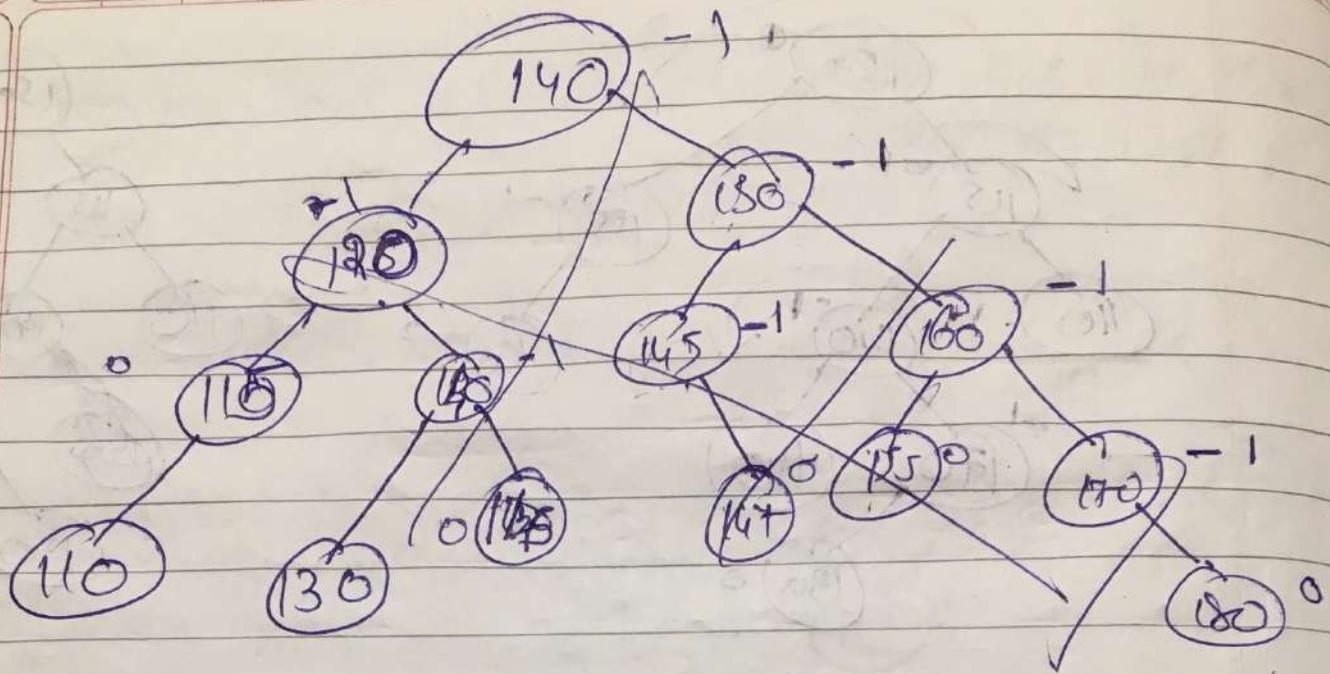




Ex: 150, 155, 160, 115, 110, 140, 120, 145, 130, 147,  
170, 180.







25/9/19

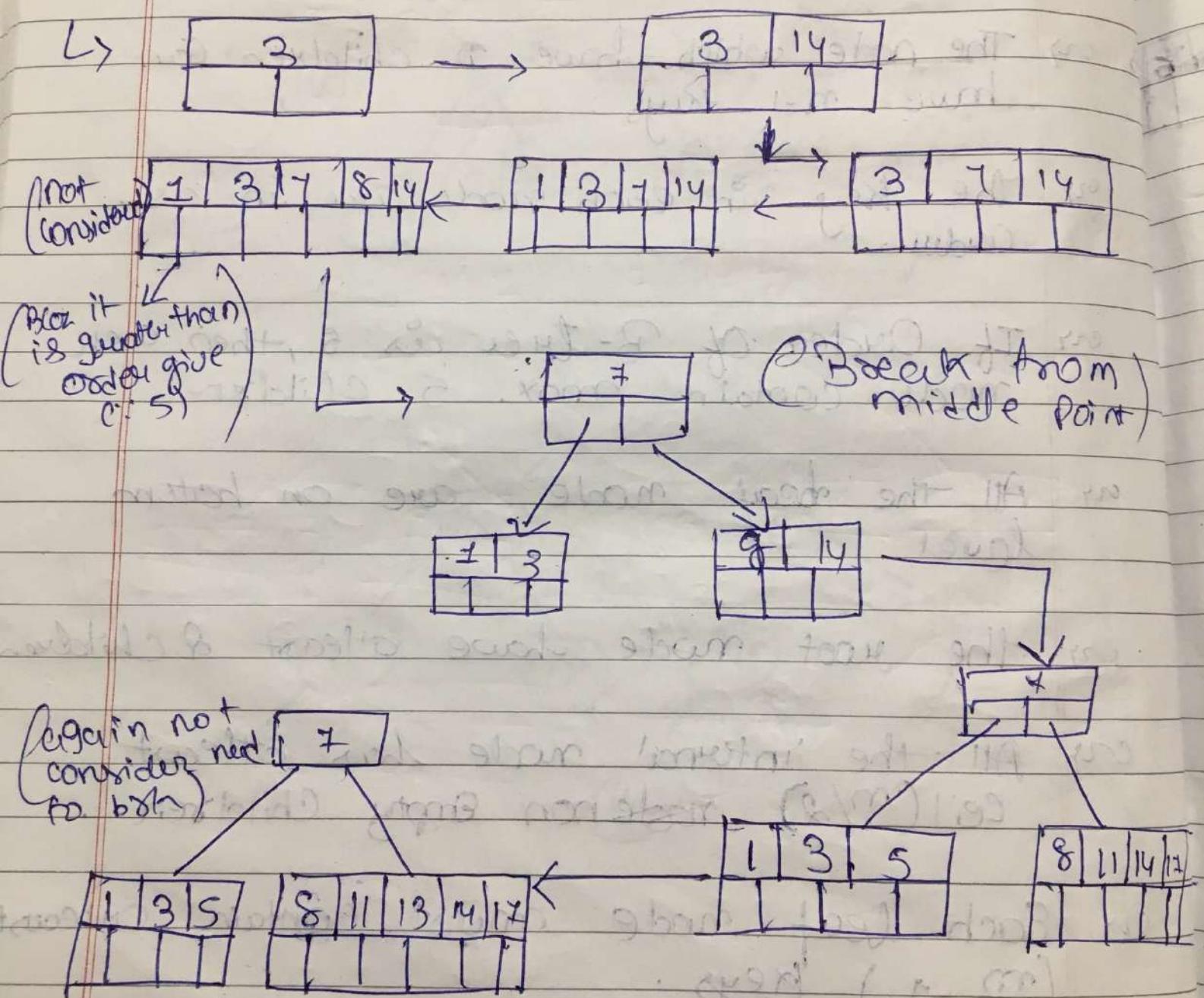
Date \_\_\_\_\_  
Page \_\_\_\_\_

## \* B-Tree (multi-way Search tree)

### Rules :-

- (i) The node which have  $n$  children can have  $n-1$  keys.
- (ii) The key in each node are in asc. Order.
- (iii) If Order of B-tree is 5, then each node contain max. 5 children.
- (iv) All the leaf node are on bottom level.
- (v) The root node have atleast 2 children.
- (vi) All the 'internal' node has atleast  $\lceil \frac{m}{2} \rceil$  non empty children.
- (vii) Each leaf node must contain atleast  $\lceil \frac{m}{2} - 1 \rceil$  keys.

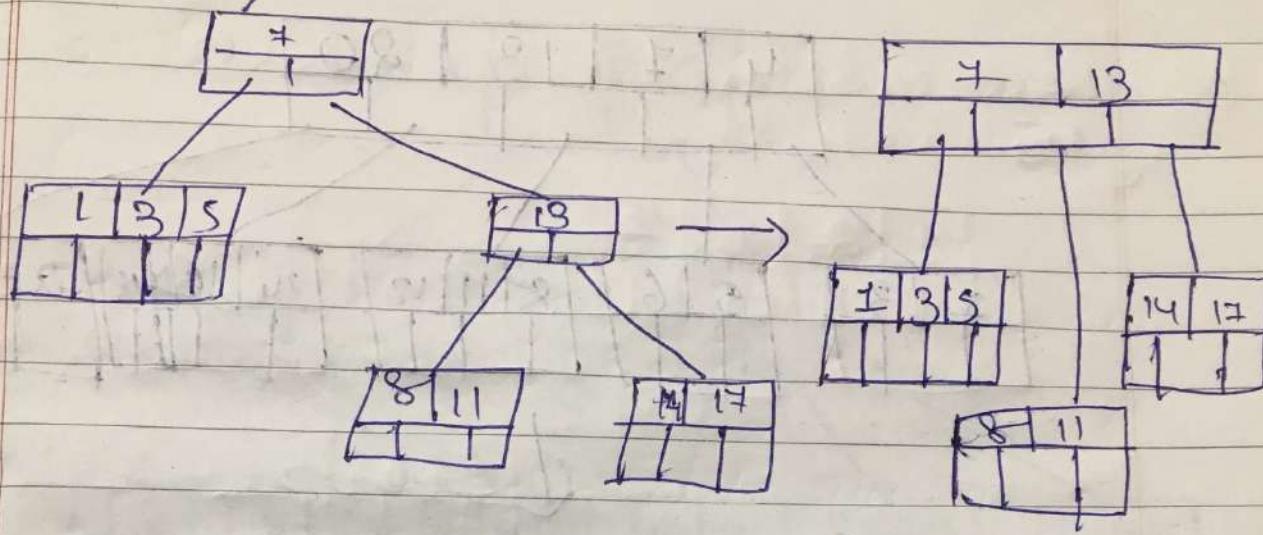
\* Create B-tree for foll. data :- (5 order B-tree)  
 3, 14, 7, 1, 8, 5, 11, 17, 13, 6, 23, 12, 20, 26,  
 4, 16, 18, 24, 19.



(for understanding  
no need to do this  
step)

Date \_\_\_\_\_

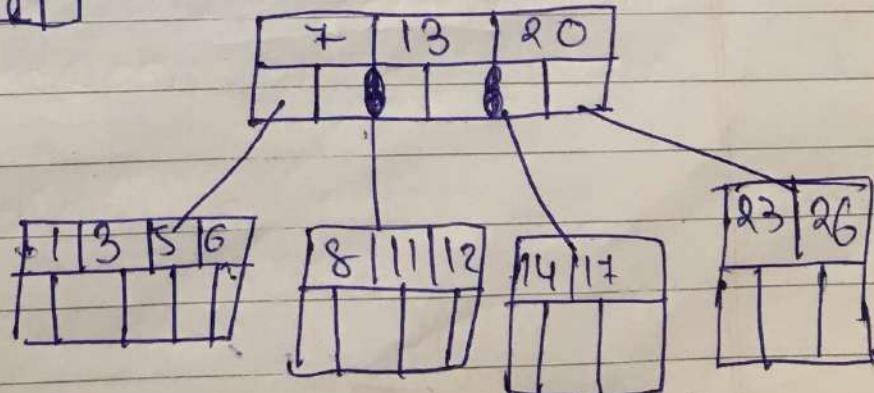
Page \_\_\_\_\_

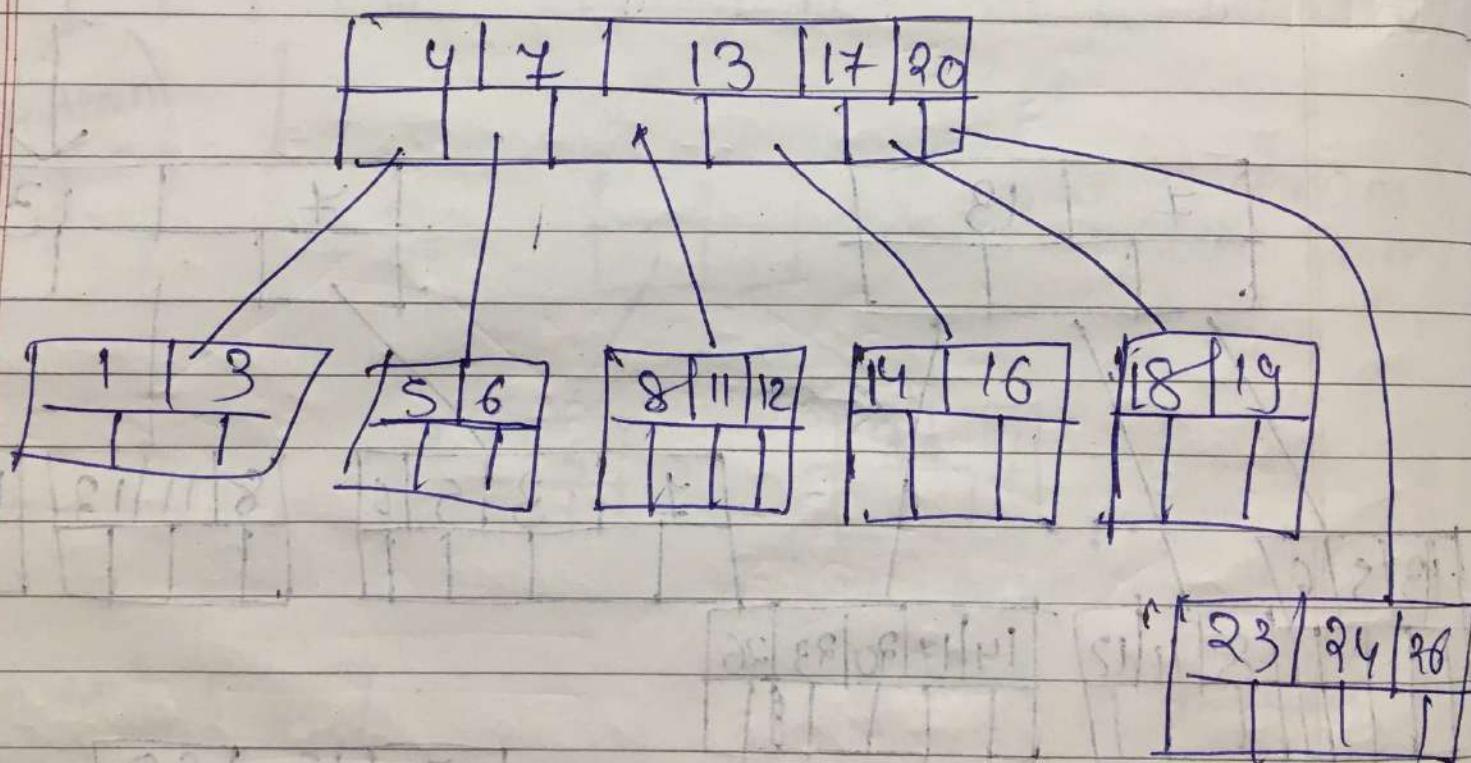
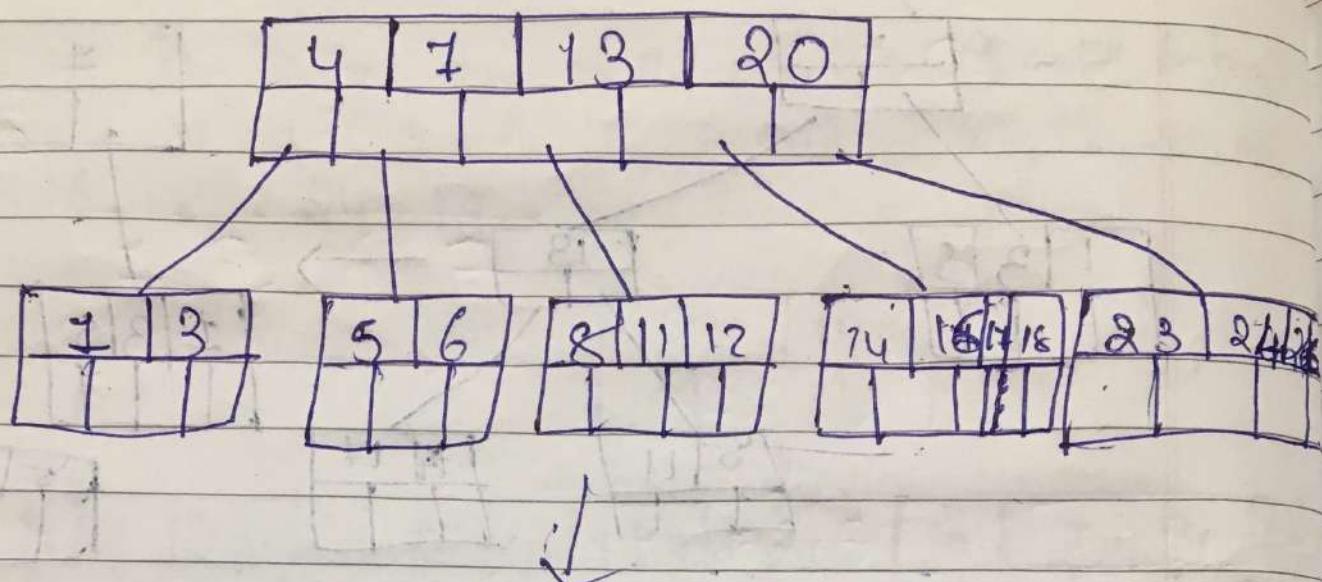


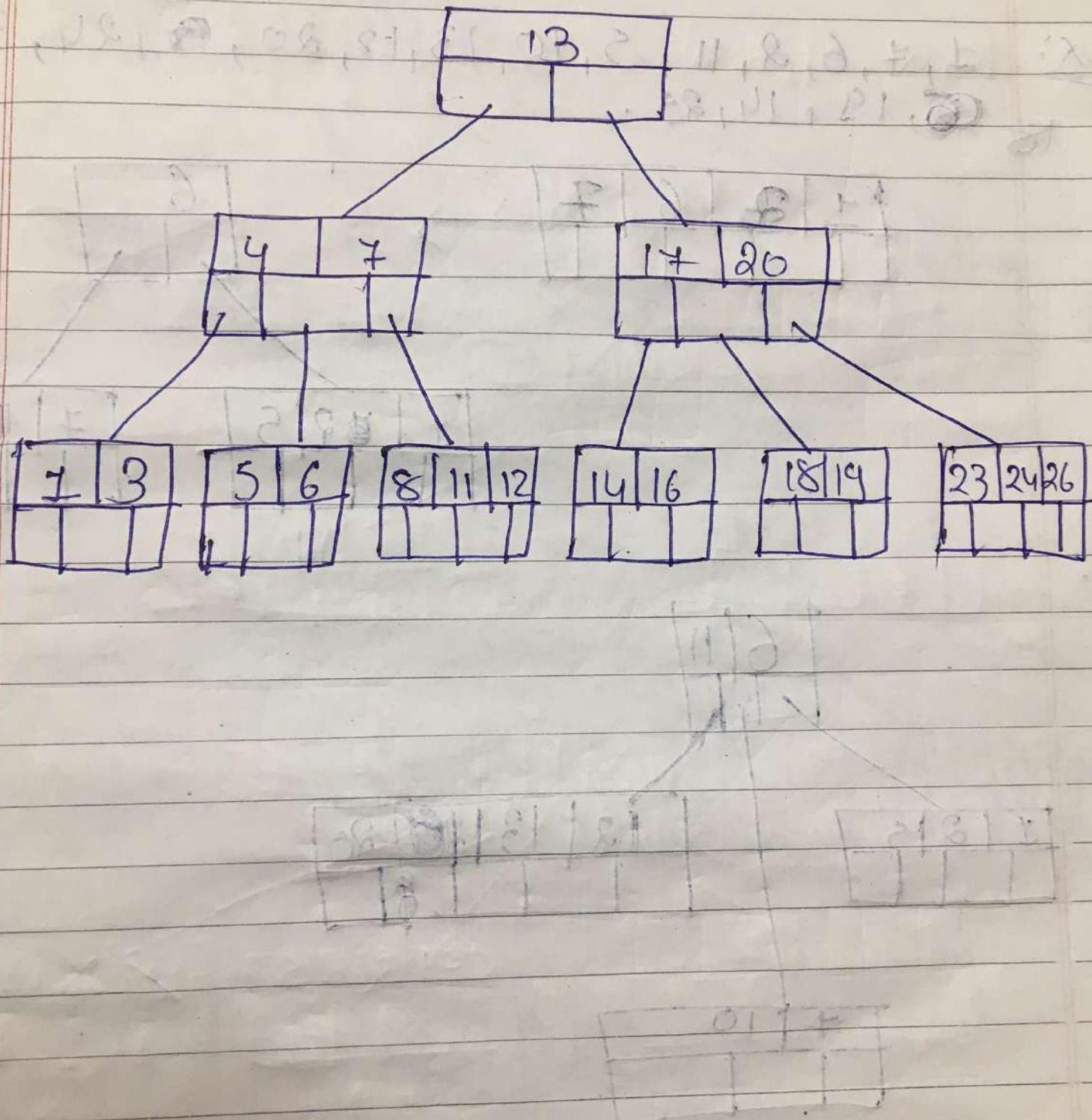
7	13	
1	3	5

7	13				
8	11	12	14	17	23

13	5	6					
8	11	12	14	17	20	23	26







Date \_\_\_\_\_  
Page \_\_\_\_\_  
(Column = 5)

Ex: 1, 4, 6, 8, 11, 5, 10, 13, 12, 20, 16, 24, 3, 4,  
18, 19, 14, 23.

16

1	2	6	7
1	2	6	7

6
1

1	2	5
1	2	5

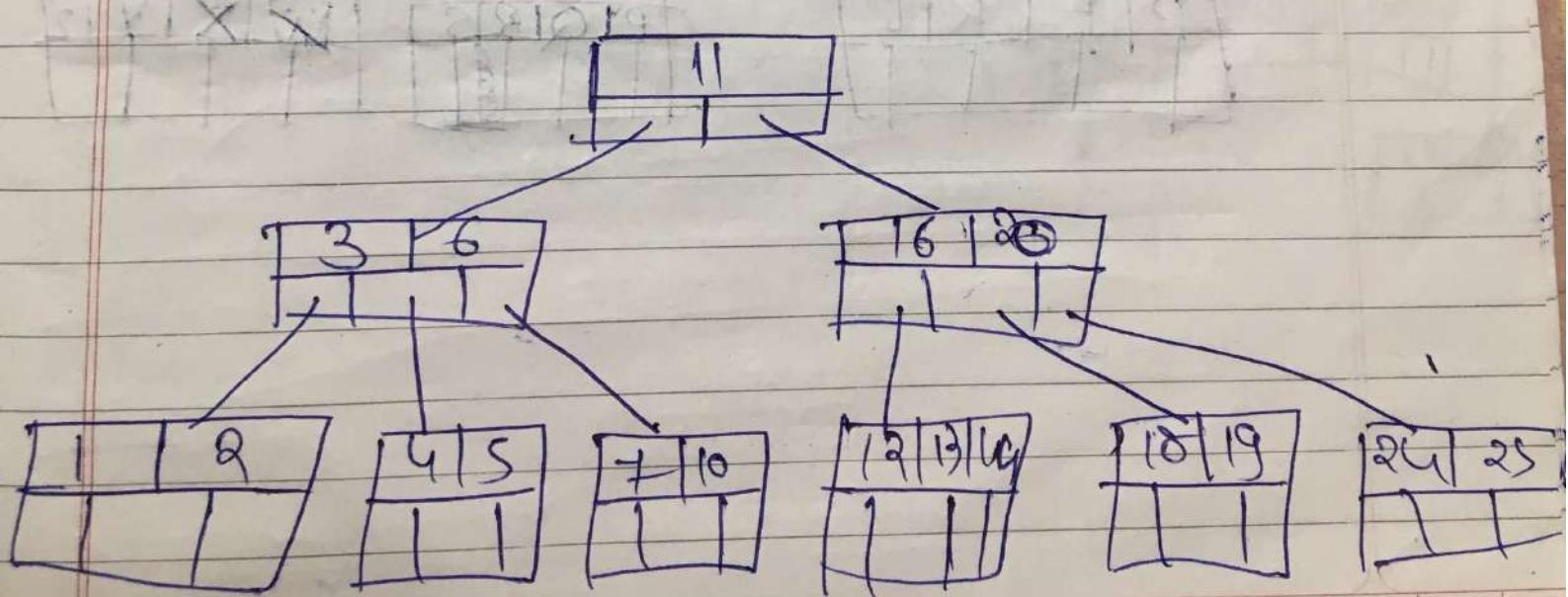
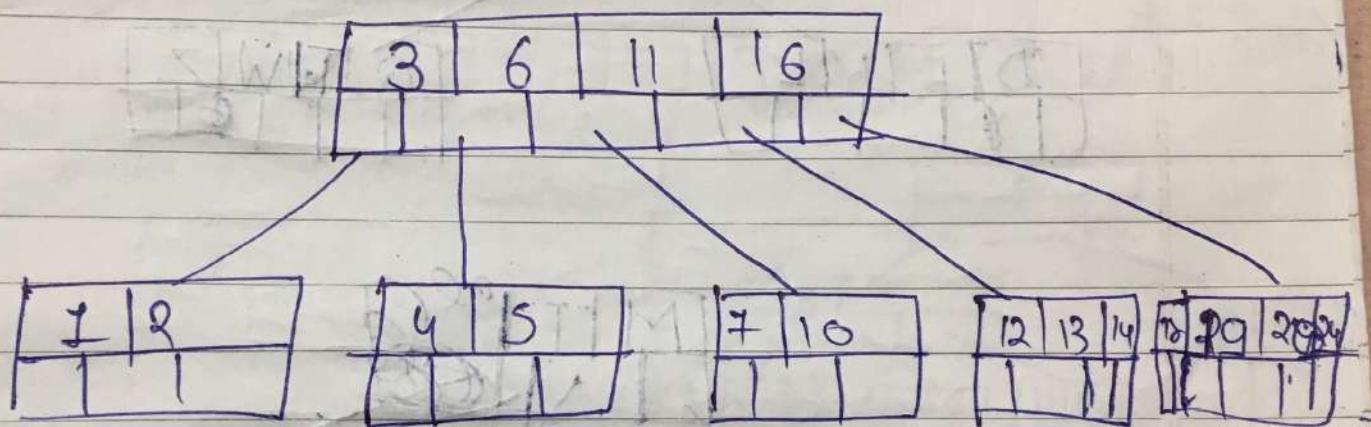
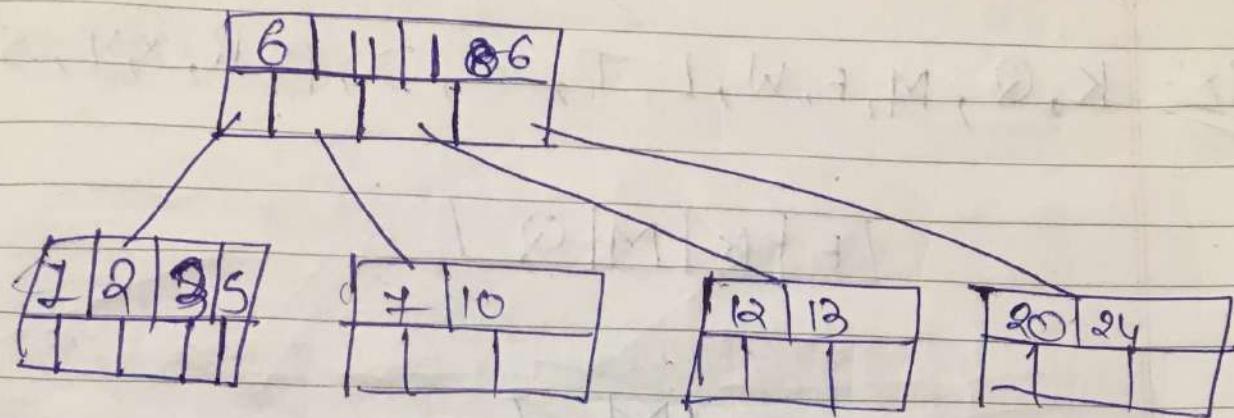
7	10	11	13
1	1	1	1

6	11
1	1

1	2	5
1	2	5

12	13	16	20
1	1	1	1

7	10
1	1



8

Date \_\_\_\_\_

Page \_\_\_\_\_

Ex: K, Q, M, F, W, L, T, Z, D, P, R, X, Y, S

F	K	M	Q
---	---	---	---

T	M
L	

D	I	F	K	L
T				

Q	T	W	Z
T			

M	I	T
L		

P	I	F	K	L
T				

P	Q	R	S
T			

X	Y	Z
T		

O	I	T
T		

S	T	U
T		

W	Y	Z
T		

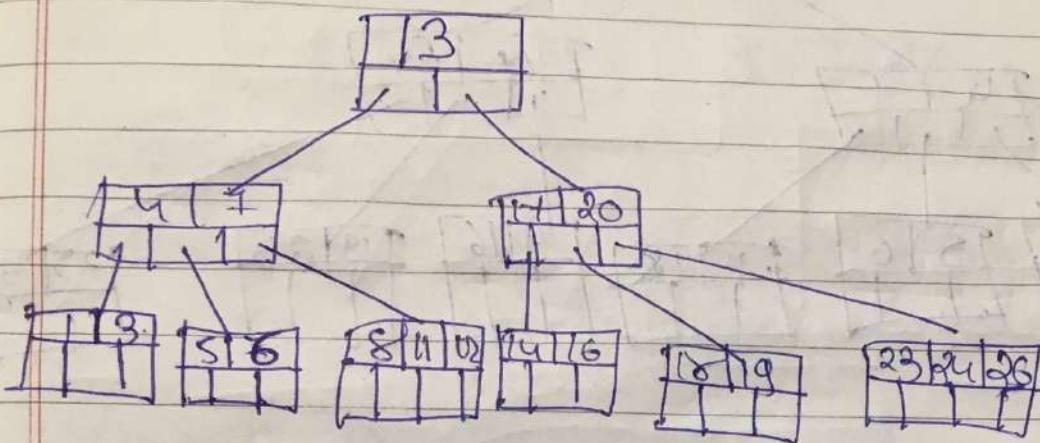
U	V	W
T		

Q	R	S
T		

2	3	4
T		

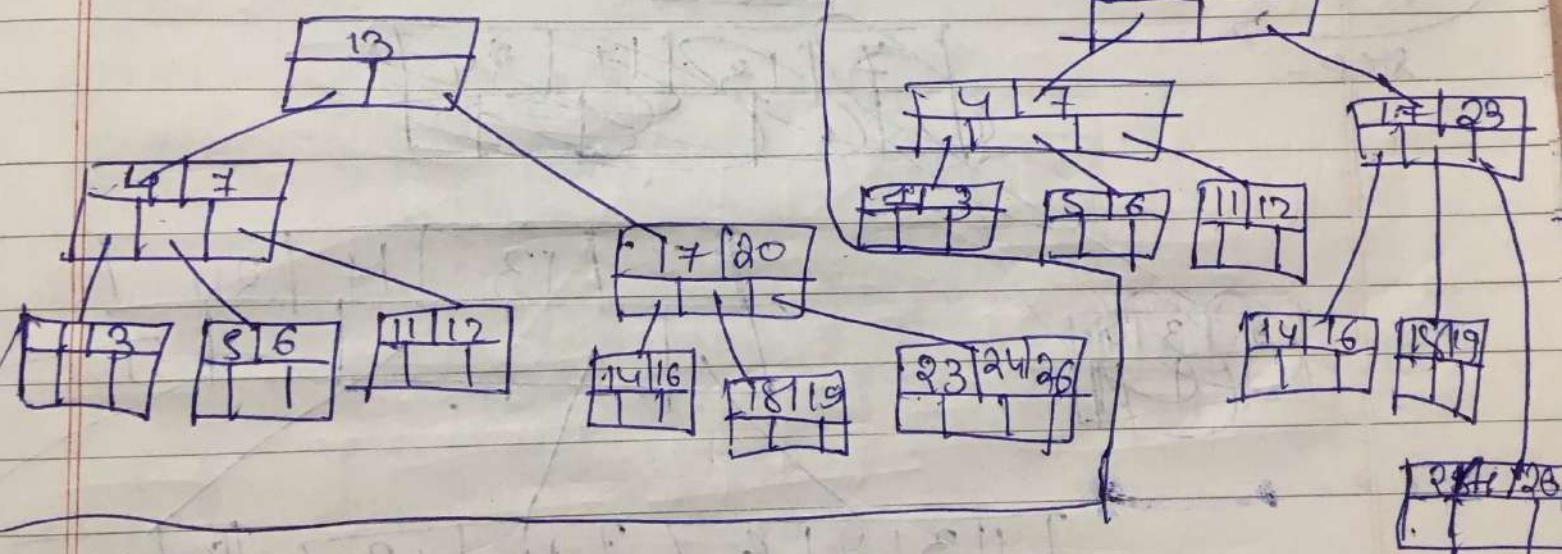
8	9	10
T		

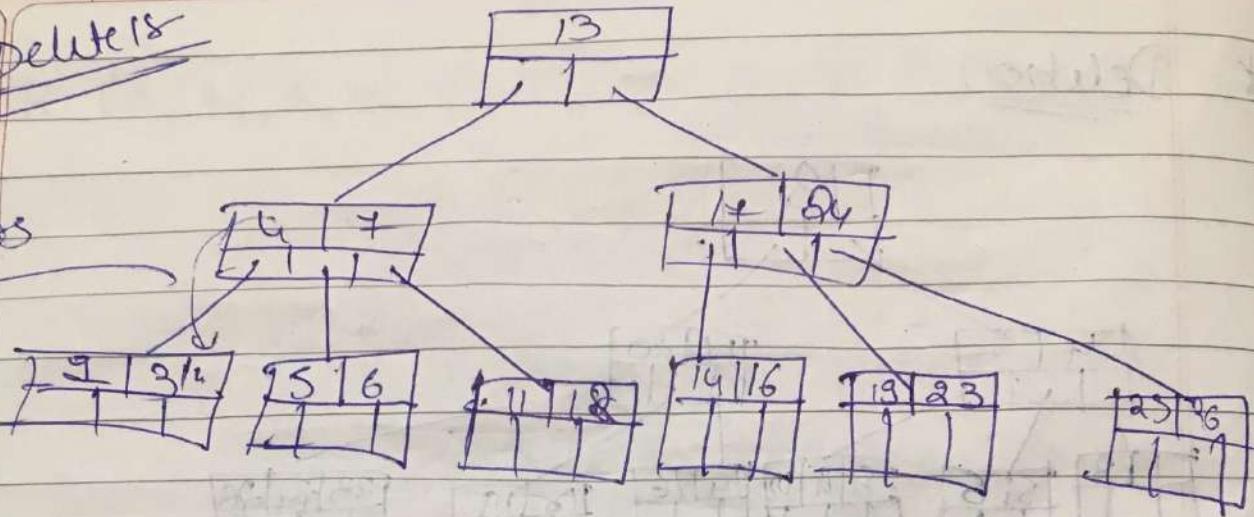
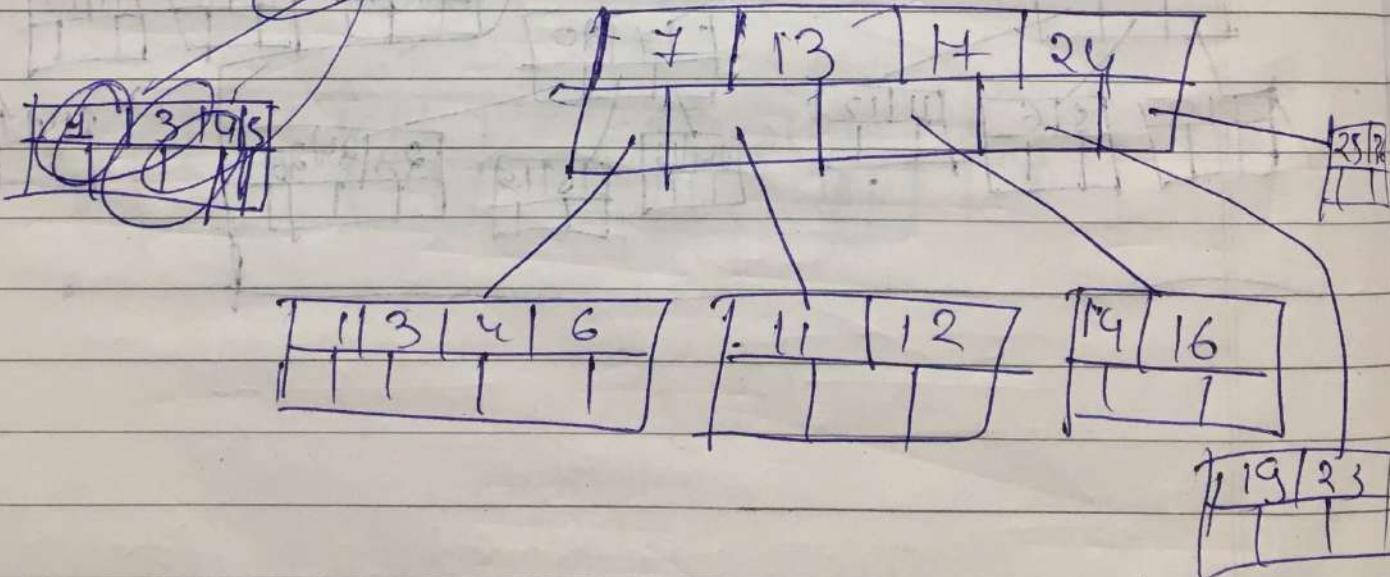
## \* Deletion



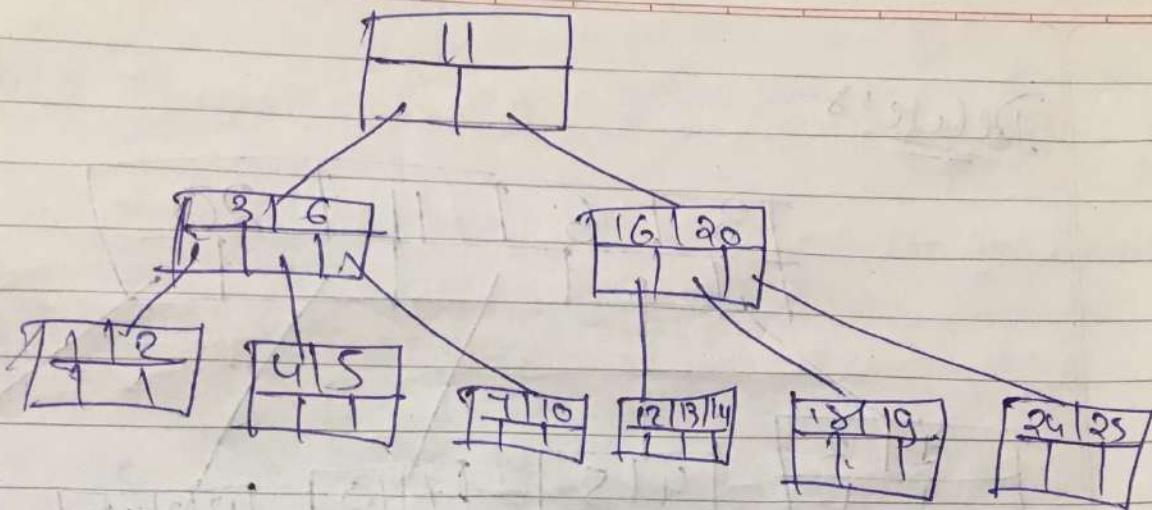
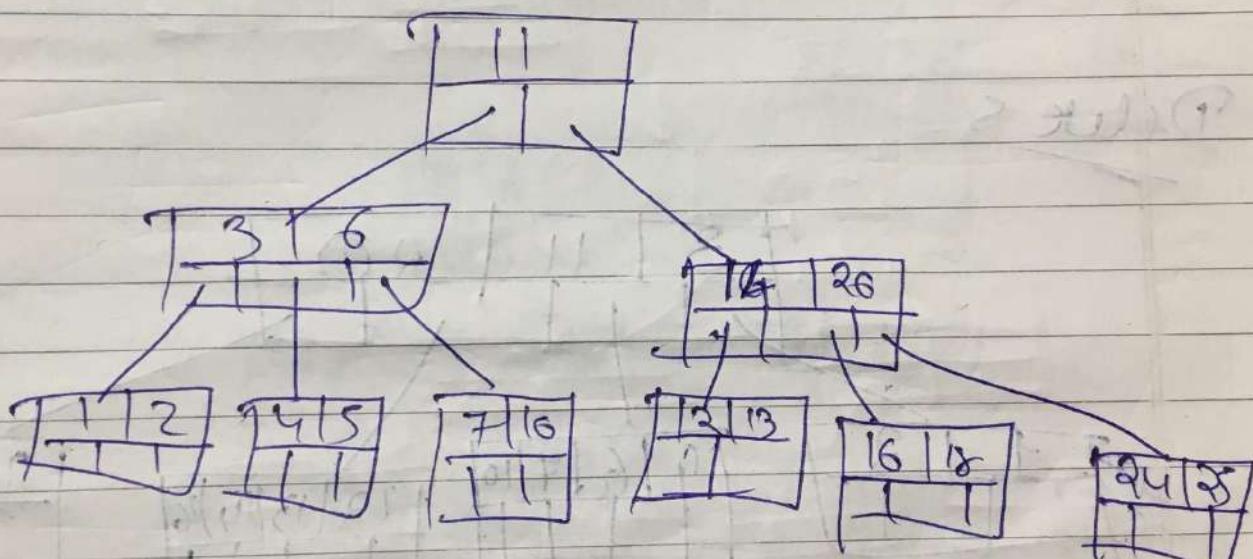
Delete 20 <sup>(Delete successful)</sup>

## Delete 8

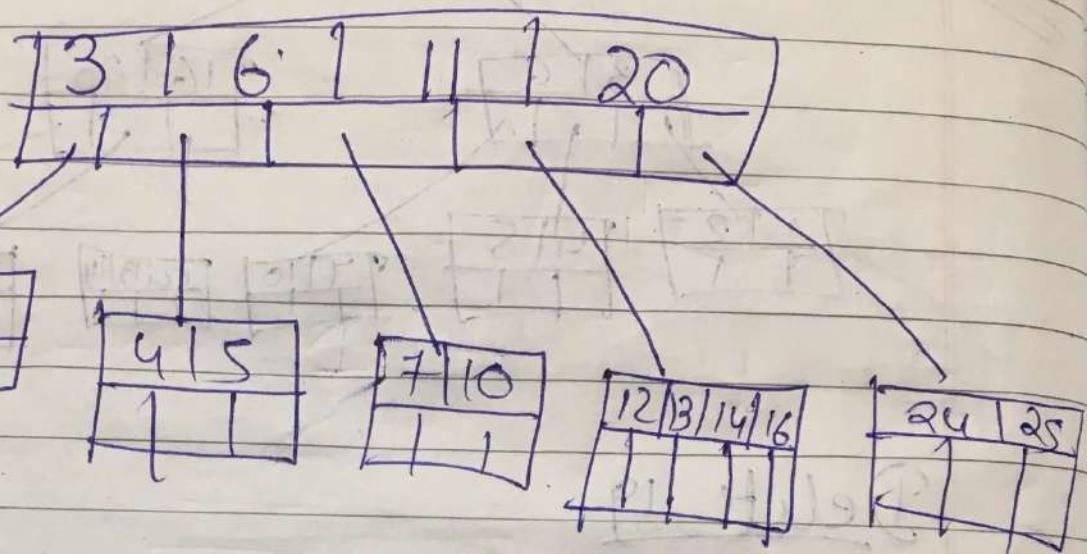


~~Delete 18~~~~Delete 5~~then final  
step -

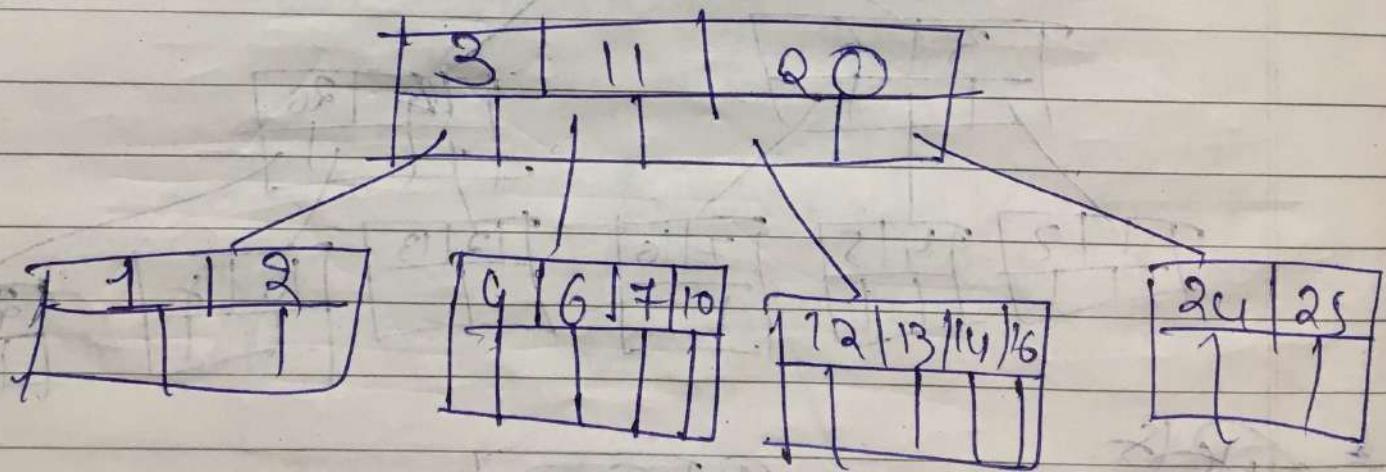
Ex:

Delete 19~~delete 18~~~~3 6 10~~

Delete 18



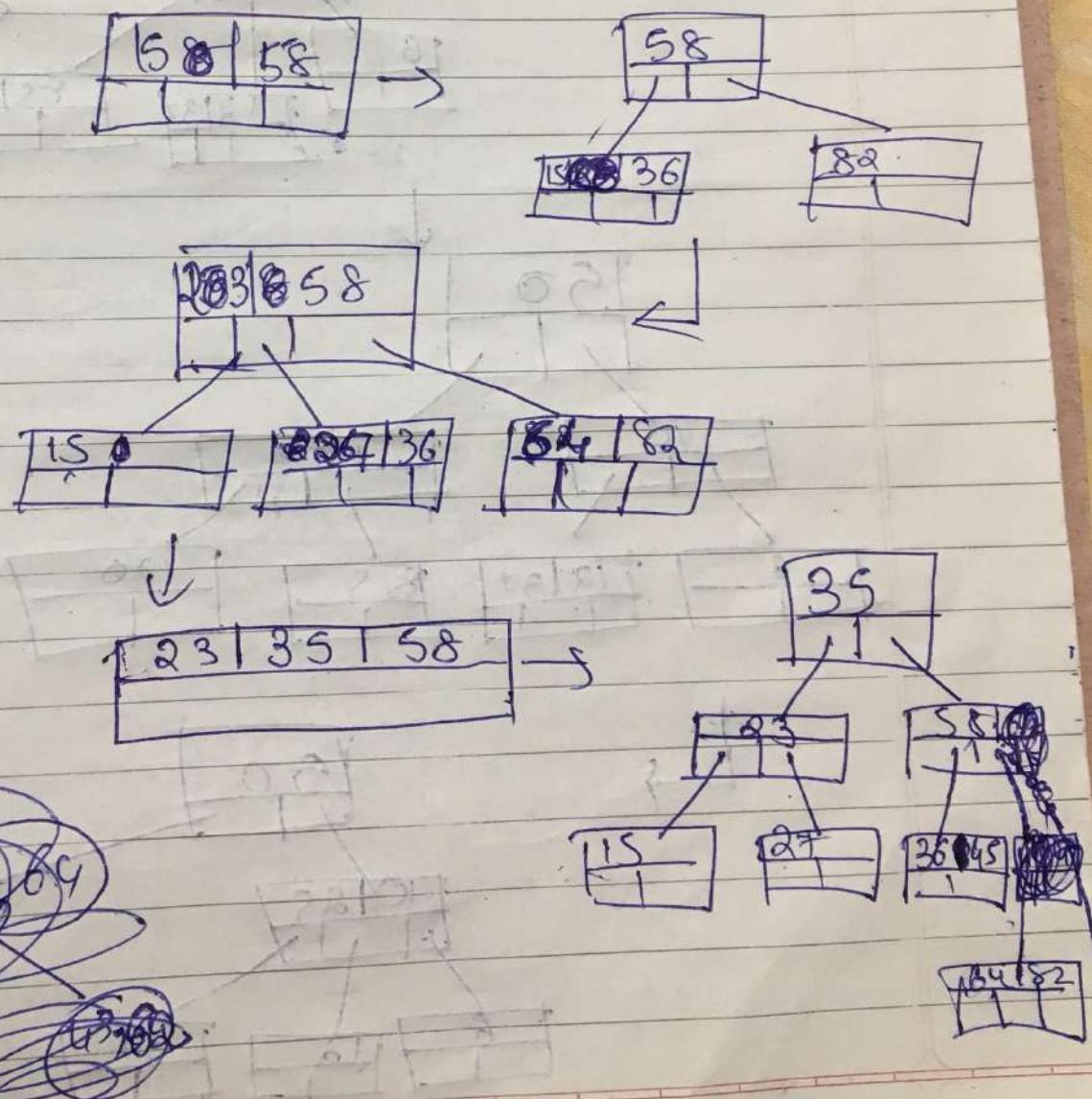
Delete 5



\* 2-3 Tree → in this order is (3)  
[otherwise same as B-tree]

→ It is type of B-tree in which every node has either 2 or 3 children. Except leaf nodes.

Ex: 58, 15, 82, 36, 23, 64, 27, 35, 45.



Ex: (12, 50, 85, 6, 10, 37, 100, 120, 25, 70)

