24/4/21 HW-6- AVL, 2-4 Trees, Hashing 19453

1.) @ Order:- 15, 20, 24, 10, 13, 7, 30, 36, 25

Insert = 15;

Insert = 20;

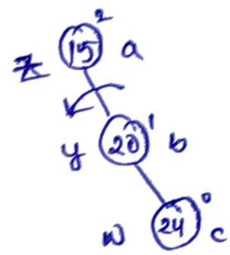
(20 > 15)

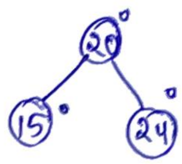
Insert = 24

(24 > 20)

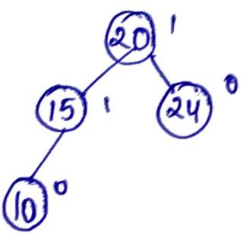
Inorder = 15, 20, 24
a b c

here b = y;
single rotation

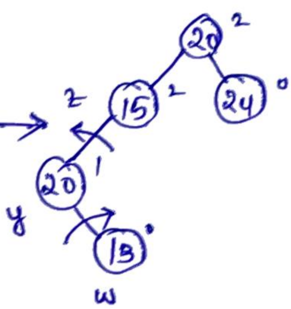




Insert = 10;

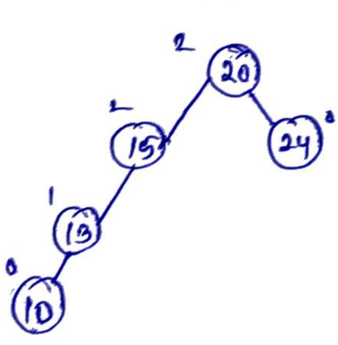


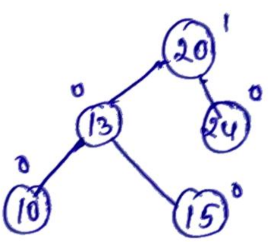
Insert = 13;



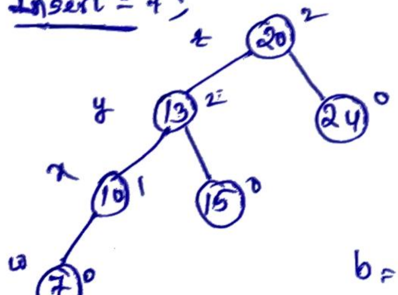
Inorder = 10, 13, 15

Double rotation



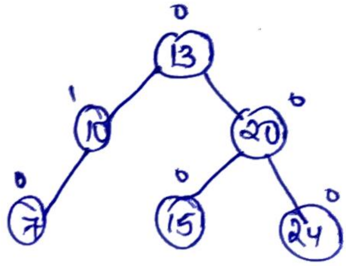


Insert = 7;

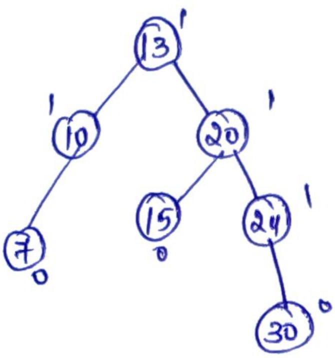


inorder:
7 10 30 20
a b c

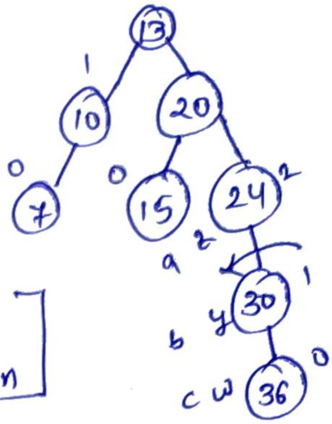
b = y; Single rotation;



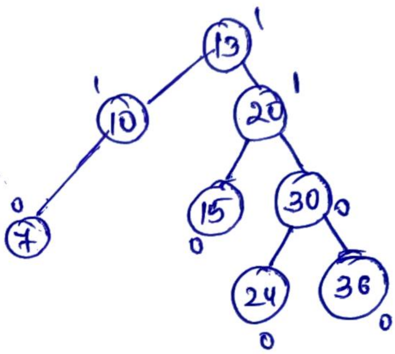
Insert = 30;



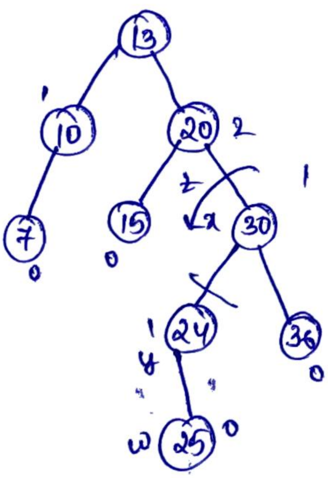
Insert = 36;



[Single rotation]



Insert = 25;

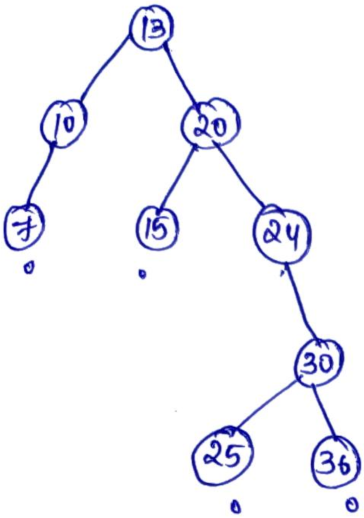


Inorder:

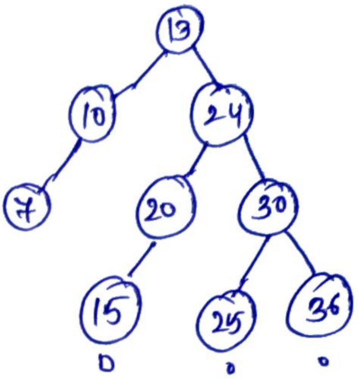
20 24 25 30

a b c

(Double rotation)

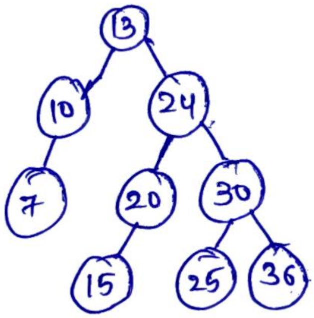


Final:-

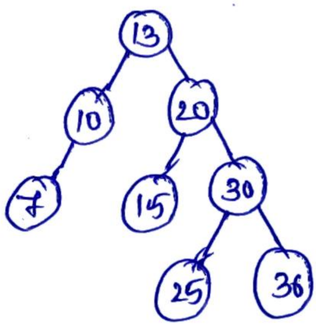


1(b))

AVL tree:

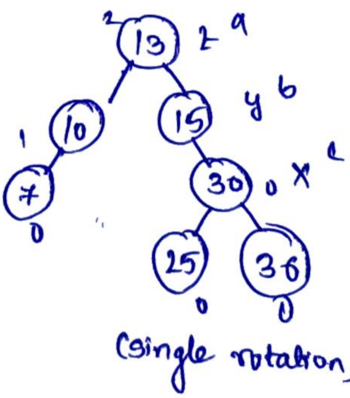


delete 24:



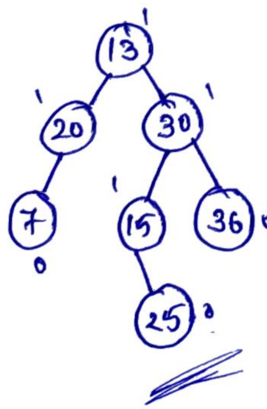
Inorder: 13, 15, 30
a b c

delete = 20:



(single rotation)

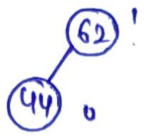
final



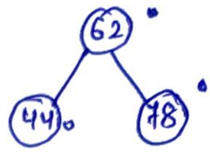
2.) Insert:- 62, 44, 78, 17, 50, 88, 48, 54 after delete key(62)

insert = 62;

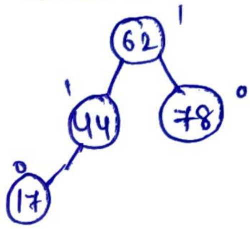
insert = 44;



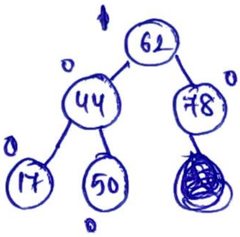
insert = 78;



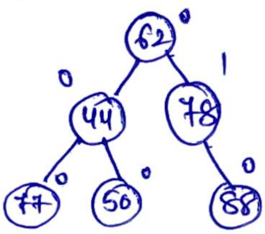
insert = 17;



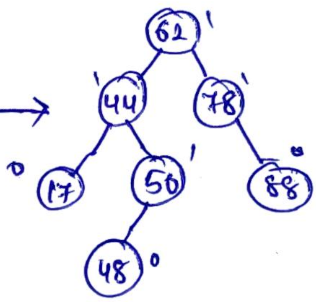
insert = 50;



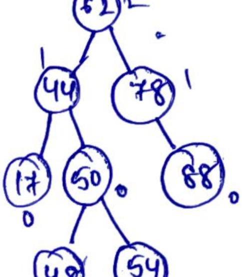
insert = 88;



insert = 48;



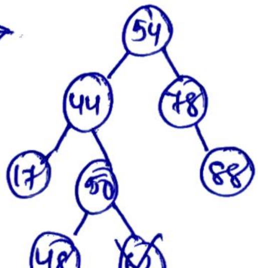
insert = 54;



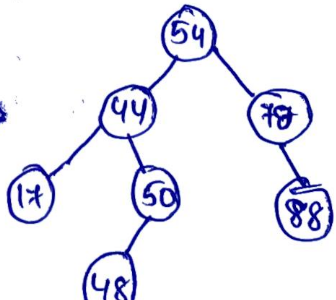
delete 62;

(i) replace 62 with 54;

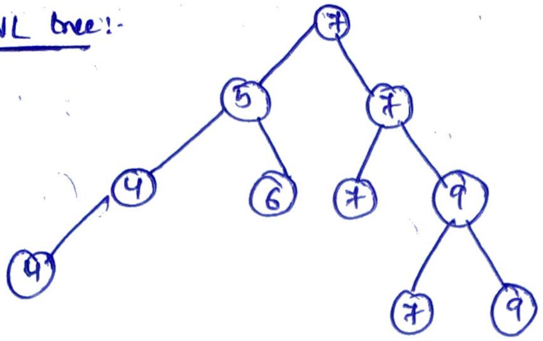
(ii) now, delete 62;



Final:

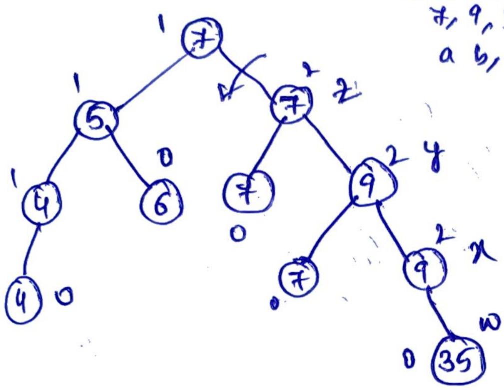


3) AVL tree :-



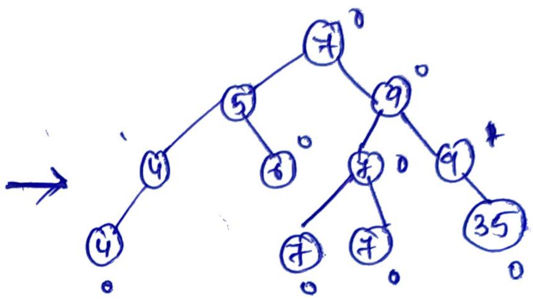
insertion of 35, 69, 100 :-

insert : 35 :

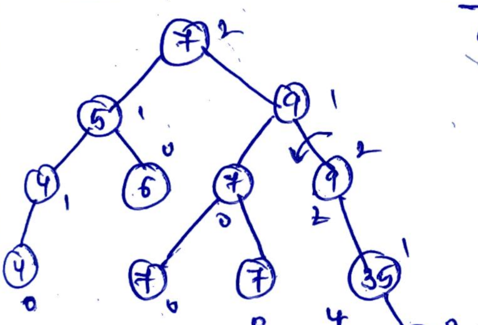


inorder :
7, 9, 35
a b, c

(single rotation)

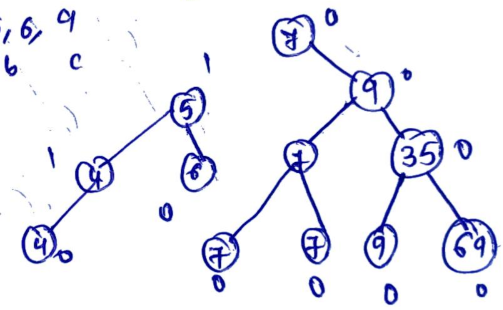


insert :- 69 :

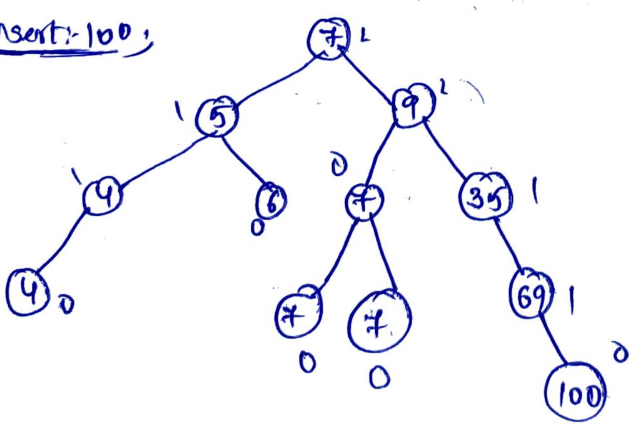


Inorder :
9, 35, 69
a b c

(single rotation)



Insert :- 100 :



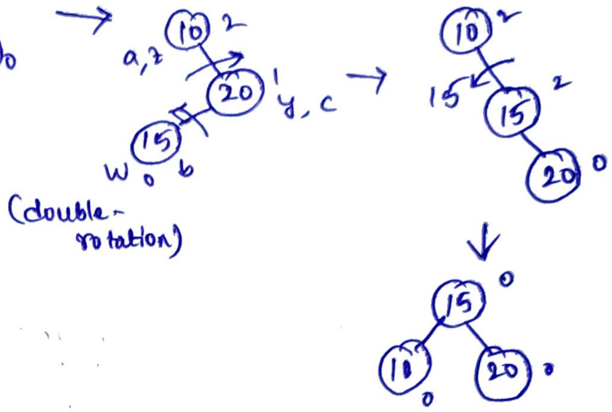
final

4.) Insertion of 10, 20, 15, 25, 30, 16, 18, 19

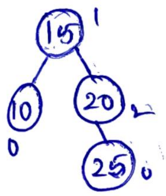
Insert-10 :

Insert = 20

Insert = 15

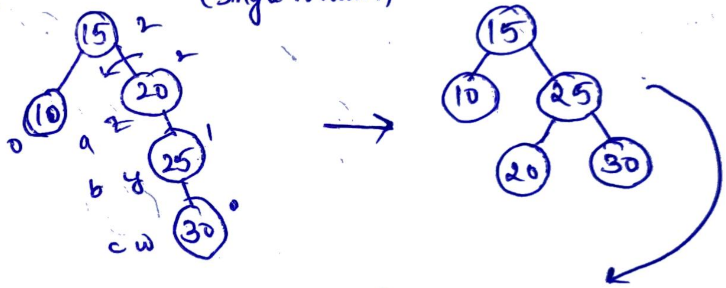


Insert = 25 ;

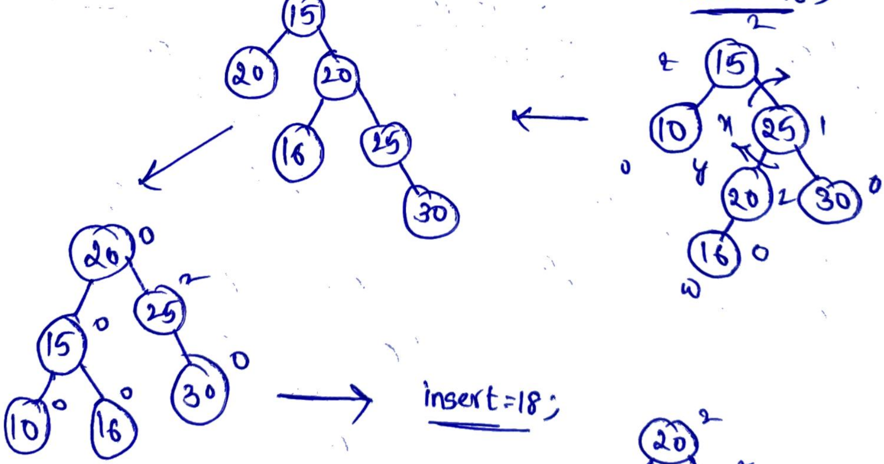


Insert = 30 ;

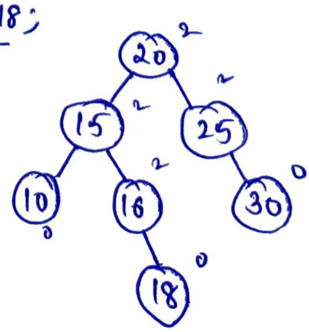
(single rotation)



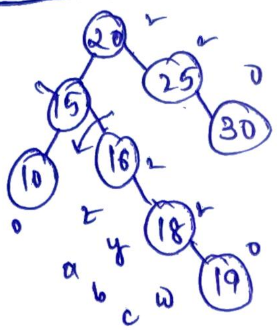
insert = 16 ;



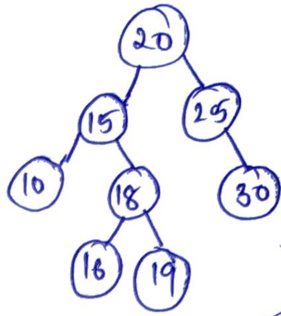
insert = 18 ;



Insert=19;



Single-rotation

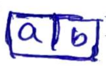


final

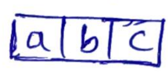
5.) K={a,b,c,d,e,f,g,h,i,j,k,l,m,n,o}

insert=a;

insert=b;

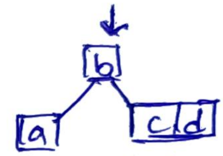


insert=c;



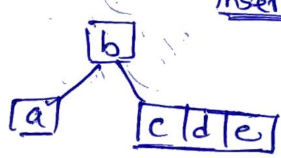
insert=d;

Overflow condition; hence split;



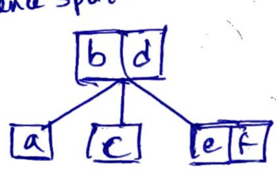
(Second element splitting)

insert=e;

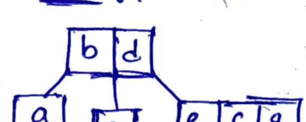


insert=f;

Overflow; hence split.

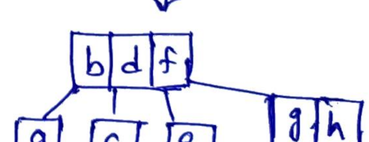


insert g;-

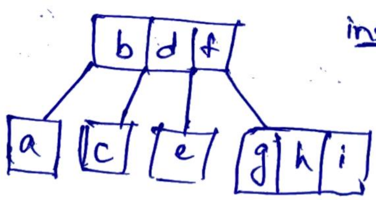


insert h;

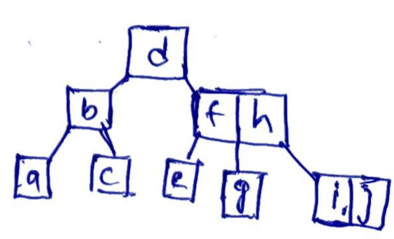
overflow hence split



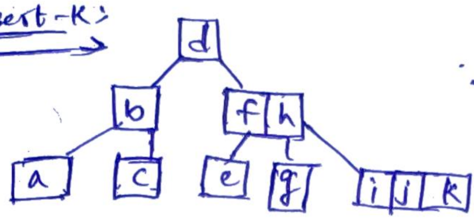
insert-i;



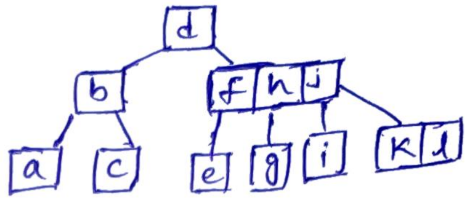
insert-j:- overflow; hence split.;



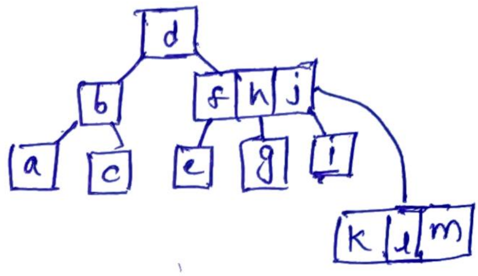
insert-k;



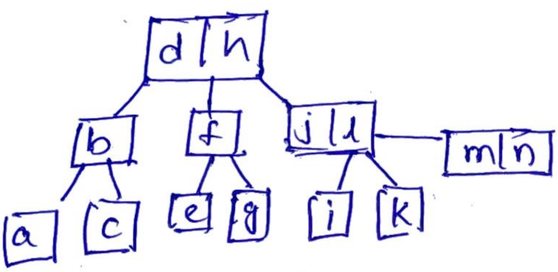
Insert-l: overflow/ hence split



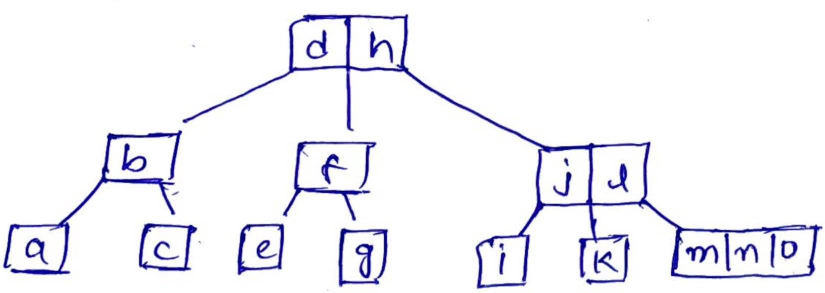
insert-m;



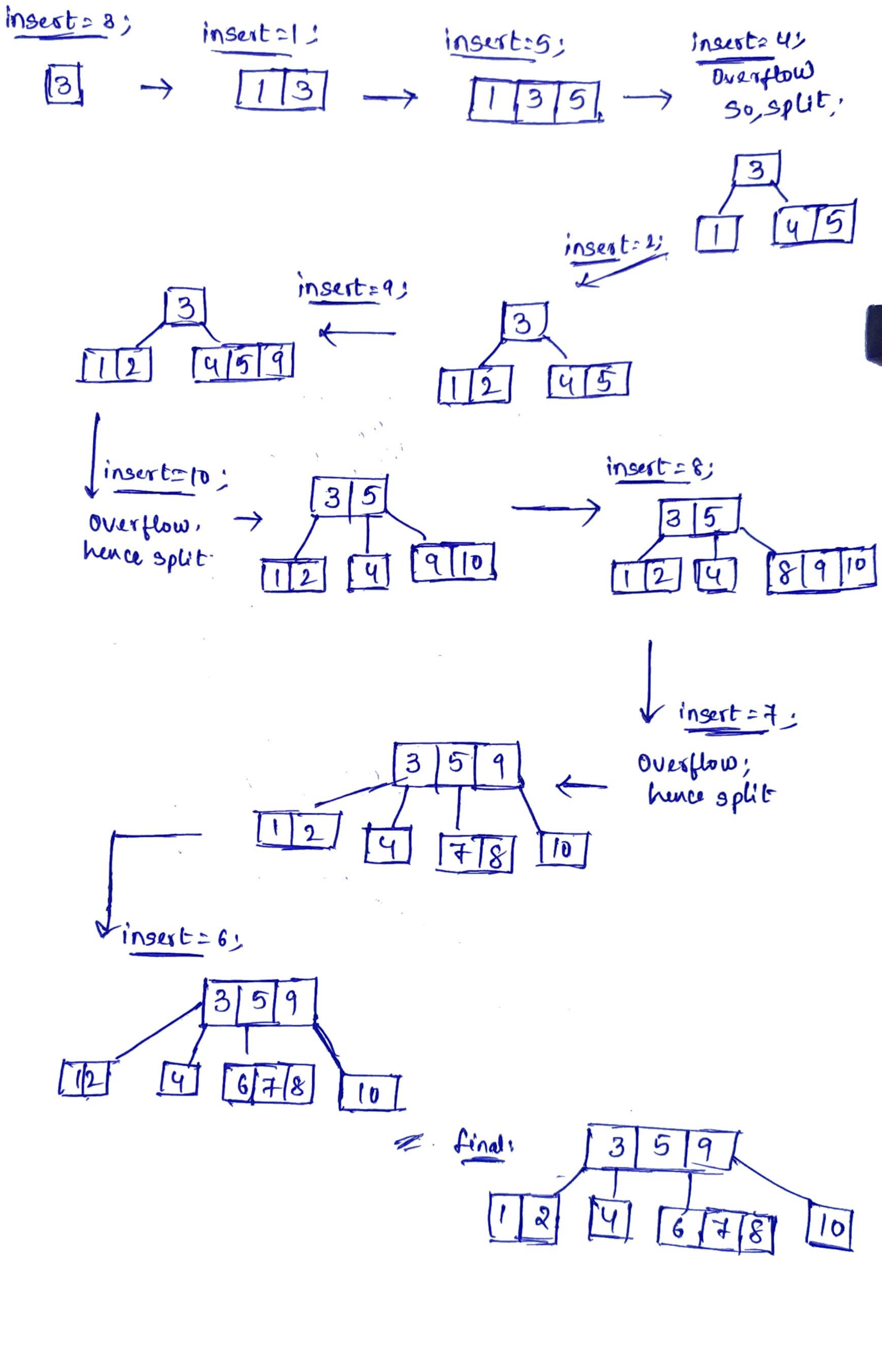
insert n;
overflow
so, split.



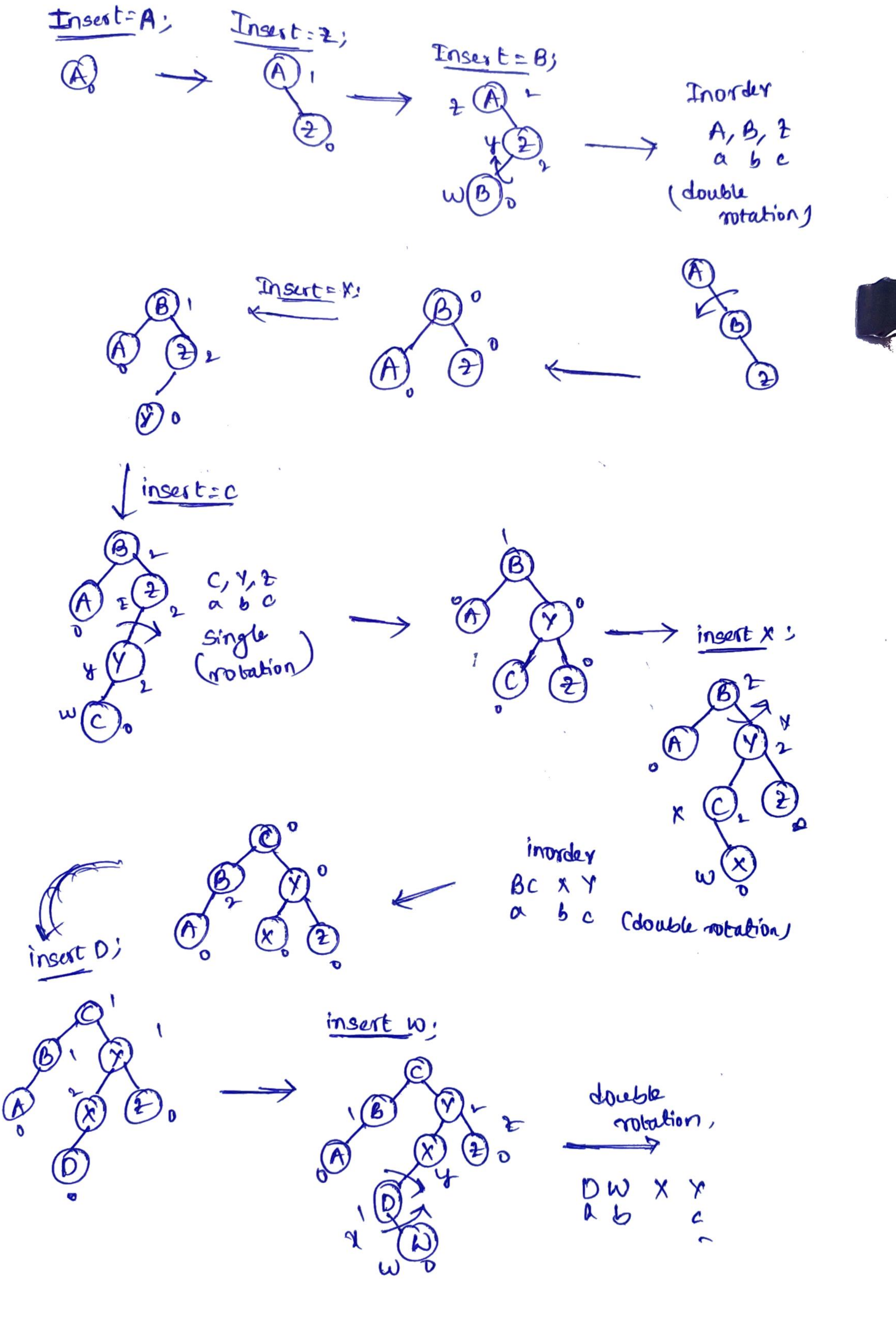
insert o;

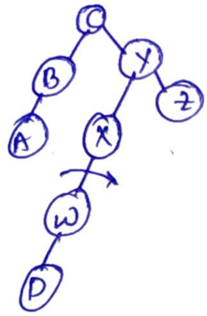


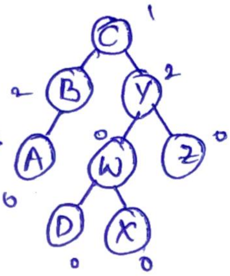
6) $K = \{3, 1, 5, 4, 2, 9, 10, 8, 7, 6\}$



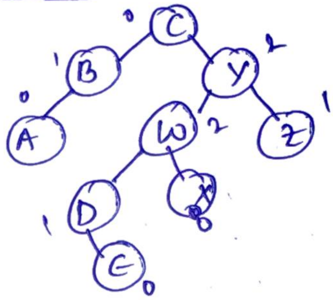
7.) (a) K = { A, Z, B, Y, C, X, D, W, E, V, F, L }

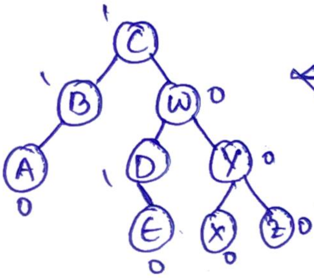






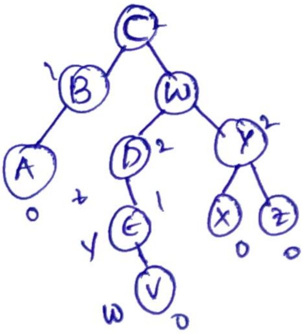
insert E;



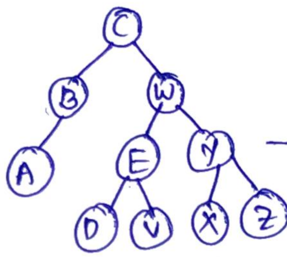


$\begin{pmatrix} D\,E & W & X \\ a & b & 0 \end{pmatrix}$ single rotation.

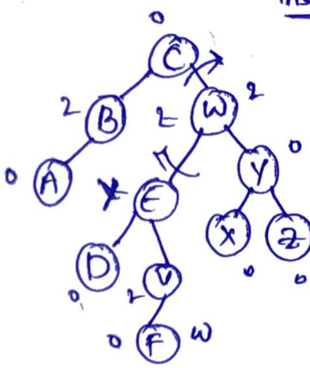
insert V;



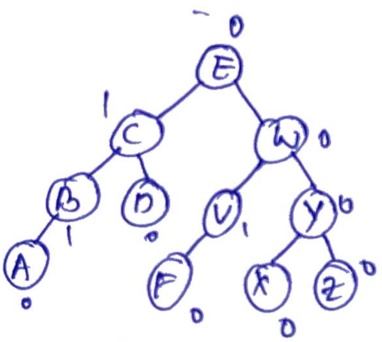
(single rotation)



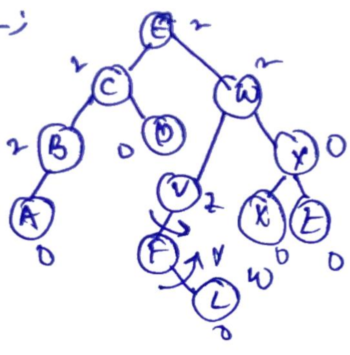
insert F;



double rotation,

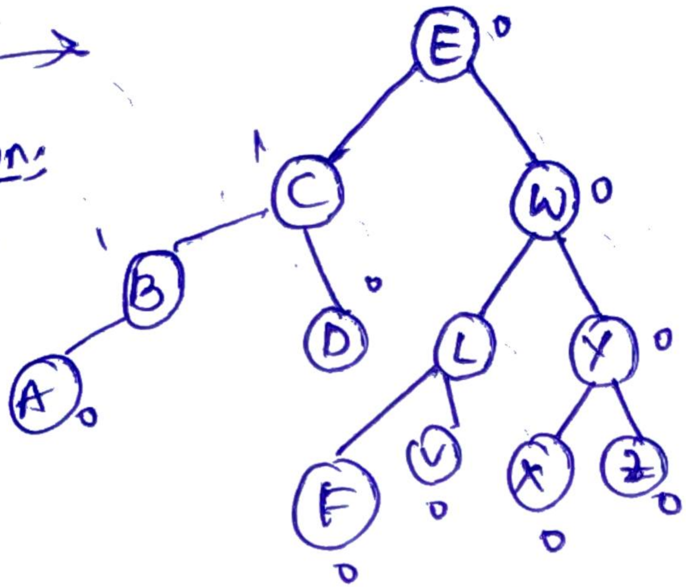


insert-L;

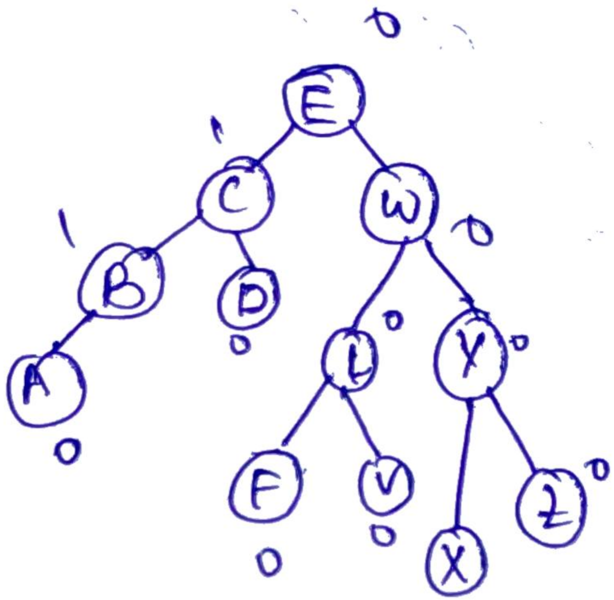


double rotation

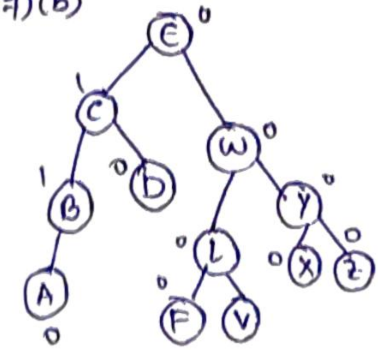
double rotation:



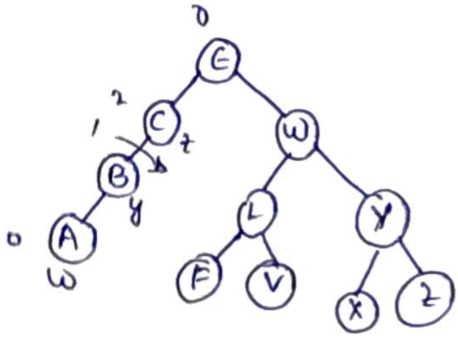
Final :-



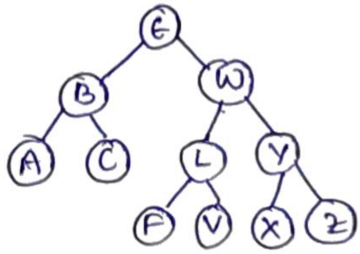
7) (b)



delete D →



A B C
a b c
(single rotation)



7 (c) :-

- AVL trees are strictly balanced.

- Here; we can find the maximum value by starting from root, and constantly travelling down the right-most branch till end.

Hence;
the time complexity for finding the max. value in an AVL tree is O(log N)
where N - no. of elements.

T. findmax() —— O(log N)

8) (a) $h(x) = x \% 7$

Size $= 7$

$I = \{2341, 4234, 2839, 430, 22, 397, 3920\}$

(a) <u>Linear probing</u>:

$2341 \% 7 = 3$

$4234 \% 7 = 6$

$2839 \% 7 = 4$

$430 \% 7 = 3$ – collision

$(430+1) \% 7 = 4$

$(430+1) \% 7 = 5$

$22 \% 7 = 1$

$397 \% 7 = 5$ — collision

$(397+1) \% 7 = 6$

$(397+2) \% 7 = 0$

$3920 \% 7 = 0$ — collision

$(3920+1) \% 7 = 1$ — collision

$(3920+2) \% 7 = 2$

6	4234
5	430
4	2839
3	2341
2	3920
1	22
0	397

(b) <u>Seperate Chaining</u>:

0	3920	
1	22	
2		
3	2341	→ 430
4	2839	
5	395	
6	4234	

→ insert 2341

→ insert 4234

→ insert 2839

→ insert 430

→ Collision occurs; add to chain

→ insert 22

→ insert 397

→ insert 3920

8(c)) double hashing :

$h = (h(x) + i \cdot h'(x)) \bmod size$

$h'(x) = (2x-1)\%7$

$h(x) = x\%7$

$2341\%7 = 3$

$4234\%7 = 6$

$2839\%7 = 4$

$430\%7 = 3$ → collision

$(430\%7)+1(7-(2\times430)-1)\%7)\%7 = 5$

$22\%7 = 1$

$397\%7 = 5$ → Collision:

$(397\%7)+1(7-(2\times430)-1)\%7)\%7 = 3$

$(397\%7)+2(7-(2\times430)-1)\%7)\%7 = 1$

$(397)\%7)+3(7-(2\times430)-1)\%7)\%7 = 6$

$(397\%7)+4(7-(2\times430)-1)\%7)\%7 = 4$

$(397\%7)+5(7-(2\times430)-1)\%7)\%7 = 2$

$3920\%7 = 0$

6	4234
5	430
4	2839
3	2341
2	397
1	22
0	3920

(a)

9) Linear probing :-

(data + i) % size.

$I = \{8, 33, 15, 26, 22\}$

size = 7

$h(K) = K \bmod 7$

6	26
5	33
4	
3	22
2	15
1	8
0	

$8 \% 7 = 1$

$33 \% 7 = 5$

$15 \% 7 = 1$ — collision

$(15 + 1) \% 7 = 2$

$26 \% 7 = 5$ — collision

$(26 + 1) \% 7 = 6$

$22 \% 7 = 1$ — collision

$(22 + 1) \% 7 = 2$ — collision.

$(22 + 2) \% 7 = 3$

9) (b)

When there is collision; i (1 to size) will be added to data to find a slot empty, the process would continue till a unique key is calculated;

$$\text{load factor} = \left(\frac{\text{no. of elements}}{\text{table size}}\right)$$

$$\simeq \frac{5}{7}$$

$$= \boxed{0.7}$$

10) (a) $h(x) = x \% 10$

quadratic = $\frac{(t \cdot + i^2)}{\text{size}}$

Size = 10

$I = \{51, 23, 73, 99, 44, 79, 89, 38\}$

$51 \% 10 = 1$

$23 \% 10 = 3$

$73 \% 10 = 3$ → Collision

$(73 + 1^2) \% 7 = 4$

$99 \% 10 = 9$

$44 \% 10 = 4$ — Collision

$(44 + 1^2) \% 7 = 5$

$79 \% 10 = 9$ — Collision

$(79 + 1^2) \% 10 = 0$

$89 \% 10 = 9$ — Collision

$(89 + 1^2) \% 10 = 0$ — Collision

$(89 + 2^2) \% 10 = 3$ — Collision

$(89 + 3^2) \% 10 = 8$

$38 \% 10 = 8$ — Collision

$(38 + 1^2) \% 7 = 9$ — Collision

$(38 + 2^2) \% 7 = 2$

9	99
8	39
7	
6	
5	44
4	73
3	23
2	38
1	51
0	79

(b) When there is collision; i^2 (1 to size) will be added to key, and the process would continue till a unique key is calculated.

$$\text{load factor} = \frac{\text{no. of elements}}{\text{size of table}}$$

$$= \frac{8}{10}$$

$$= 0.8$$

ii) Chaining :-

$h(K) = K \% 9$

$I = \{5, 28, 19, 15, 20, 33, 12, 17, 10\}$

size = 9

$5 \% 9 = 5$

$28 \% 9 = 1$

$19 \% 9 = 1$ — Chaining

$15 \% 9 = 6$

$20 \% 9 = 2$

$33 \% 9 = 6$ — Chaining

$12 \% 9 = 3$

$17 \% 9 = 8$

$10 \% 9 = 1$ — Chaining