① Assignment - 1

Infix, Prefix & Postfix notation :

1) $(x + y + z) * (A + B - C)$

→ Postfix :

$((xy+)+z) * ((AB+)-C)$

$((xy+)z+) * ((AB+)C-)$

$\boxed{xy+z+AB+C-*}$

→ Prefix :

~~((+xy)+z)~~

$((x + y) + z) * ((A + B) - C)$

$((+xy) + z) * ((+AB) - C)$

$++xyz * -+ABC$

$\boxed{*++xyz-+ABC}$

2) $(A/B - C) + (D * E)$

Postfix :

$((A/B) - C) + (D*E)$

$((AB/) - C) + (DE*)$

$(AB/C-) + (DE*)$

$\boxed{AB/C-DE*+}$

(2)

Prefix:

((A/B)-C) + (D*E)
((/AB) -C) + (*DE)
(-/ABC) + (*DE)
$\boxed{+-/ABC*DE}$

3) ((((A+B)+(C/D))) * (E - (F*G)))

Postfix:

((((AB+)+(CD/))) * (E - (FG*)))
((AB+CD/+) * (EFG*-))

$\boxed{AB+CD/+EFG*-*}$

Prefix:

((((C+AB)+(/CD))) * (E - (*FG)))
(++AB/CD) * (-E*FG

$\boxed{*++AB/CD-E*FG}$

4) ((x*y)*(z*w)) - (A+B*C-D)
(xy* *z*w) - (A+B*C-D)
Postfix: ~~((xy*)*(z*w)) - (A+(B*)-D)~~
(xy*z*w* - (AB*C+D-)
~~xy*zw**ABC*+D-~~
xy*z*w* - (A+BC*-D
xy*z*w* - ABC*+D-
xy*z*w* ABC*+D

③

Prefix:

$***xyzw - (A + *BC - D)$

$***xyzw - - + A*BCD$

$-***xyzw - + A*BCD$

5) $A/B/C + D*E - f$

postfix: $AB/C/ + DE* - F$

$AB/C/DE*+F -$

Prefix: $//ABC + D*E - F$

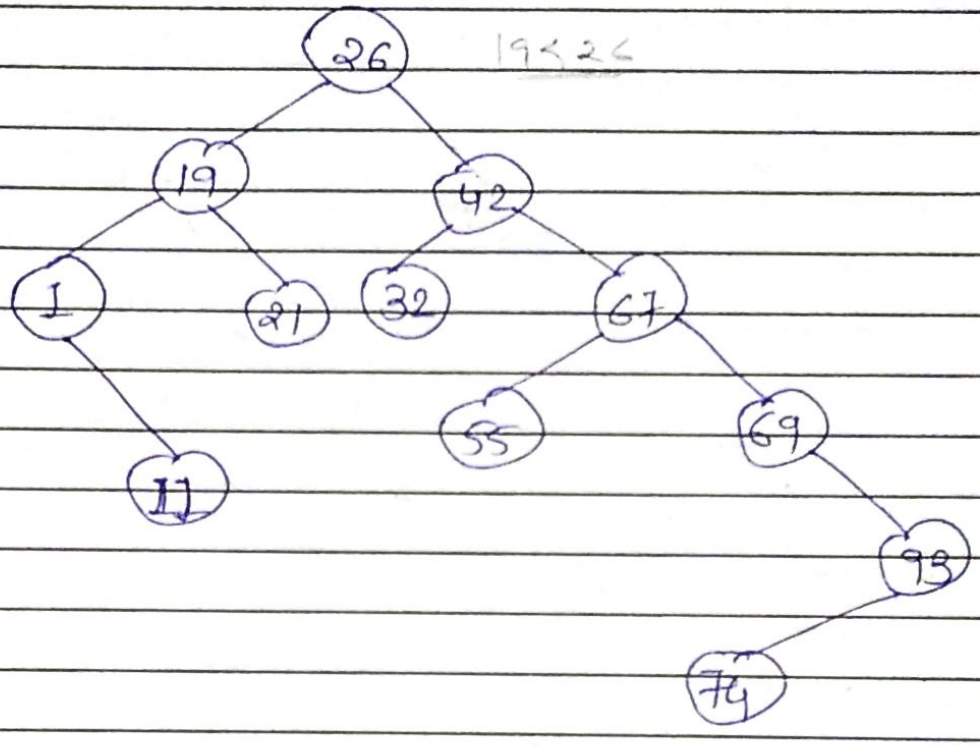
$//ABC + *DE - F$

$- + //ABC*DEf$

(4)

Q.2 Create BST for following data & perform all 3 ~~operation~~ Traversals on resultant tree.

1.) 26, 19, 42, 67, 21, 1, 69, 55, 93, 74, 32, 11



* In order Traversals: (L Root R)

→ 1, 11, 19, 21, 26, 32, 42, 55, 67, 69, 74, 93.

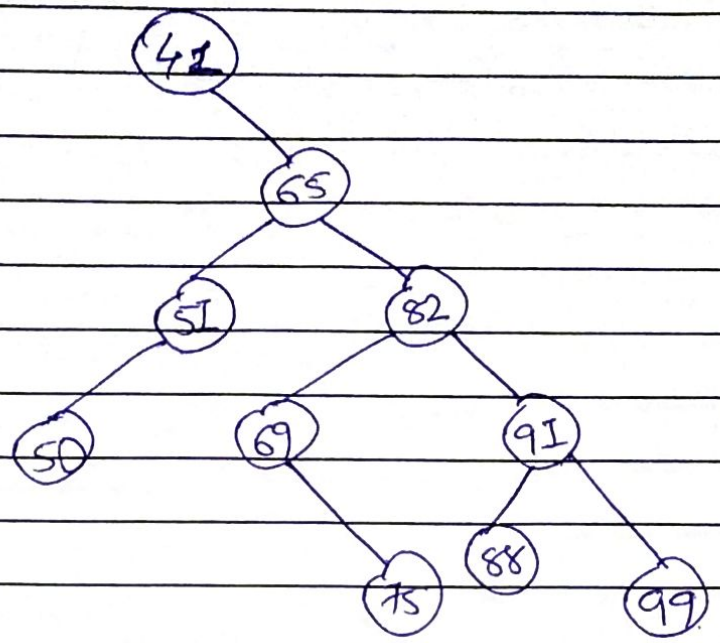
* Pre order Traversals: (Root L R)

→ 26, 19, 1, 11, 21, 42, 32, 67, 55, 69, 93, 74

* Post order Traversals: (L R Root)

→ 11, 1, 21, 19, ~~74, 93, 69,~~ 32, 55, 74, 93, 69, 67, 42, 26

2) 41, 65, 82, 91, 51, 50, 69, 75, 88, 99.



* In order: (L Root R)

→ 41, 50, 51, 65, 69, 75, 82, 88, 91, 99.

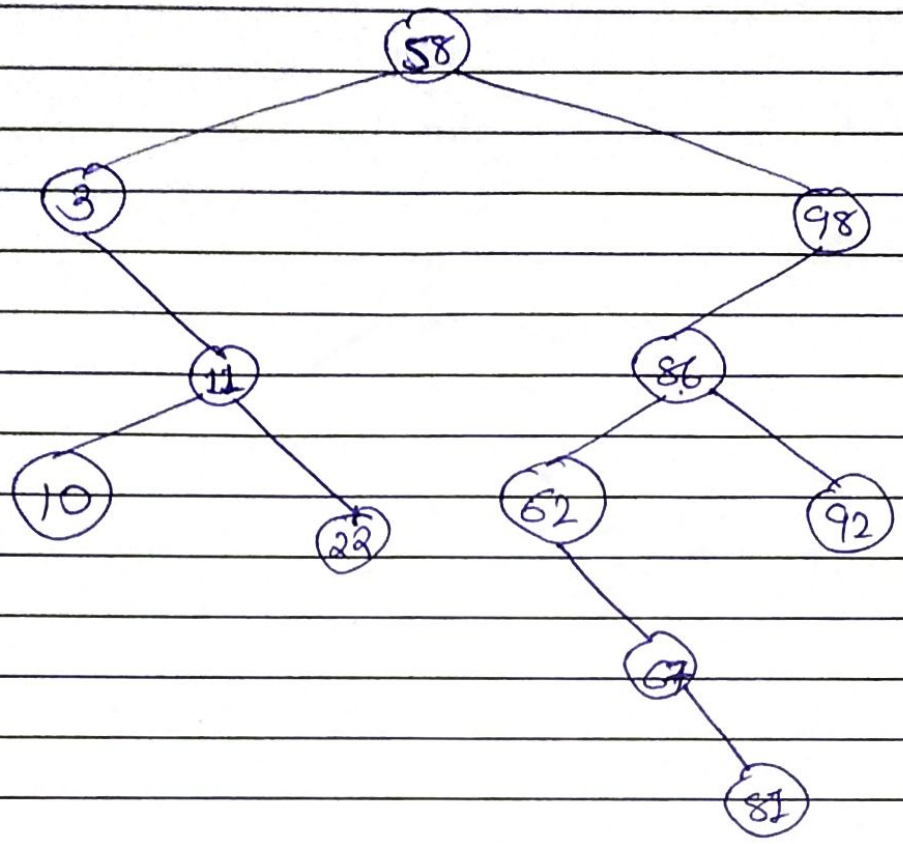
* Pre order: (Root L R)

→ 41, 65, 51, 50, 82, 69, 75, 91, 88, 99

* Post order: (L R Root)

→ 50, 51, 75, 69, 88, 99, 91, 82, 65, 41

3) 58, 3, 98, 11, 86, 22, 62, 67, 81, 92, 10



* In order : (L Root R)

→ 3, 10, 11, 22, 58, 62, 67, 81, 86, 92, 98

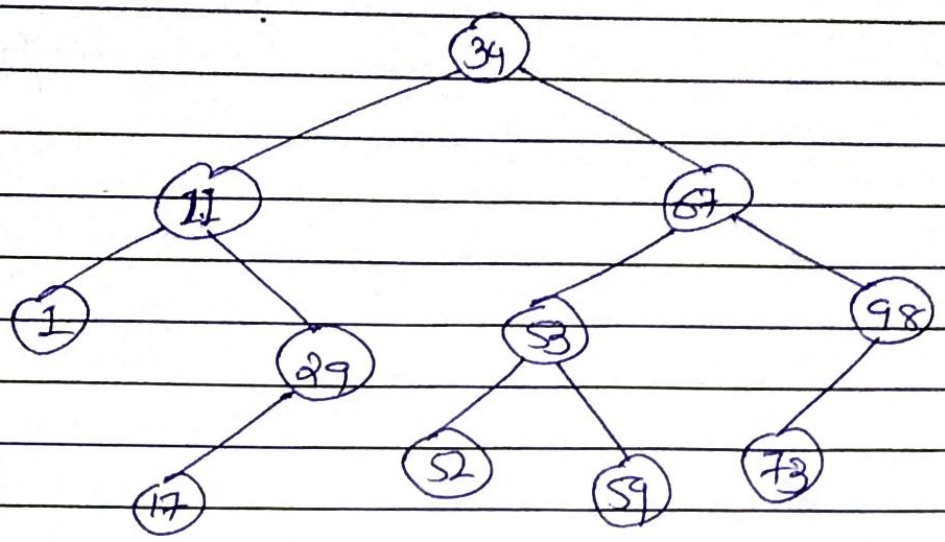
* Pre order : (Root L R)

→ 58, 3, 11, 10, 22, 98, 86, 62, 67, 81, 92

* Post order : (L R Root)

→ 10, 22, 11, 3, 81, 67, 62, 92, 86, 98, 58

Q.4 34, 11, 67, 98, 1, 73, 53, 29, 52, 17, 59



* In order: (L Root R)

→ 1, 11, 17, 29, 34, 52, 53, 59, 67, 73, 98

* Pre order: (Root L R)

→ 34, 11, 1, 29, 17, 67, 53, 52, 59, 98, 73

* Post order: (L R Root)

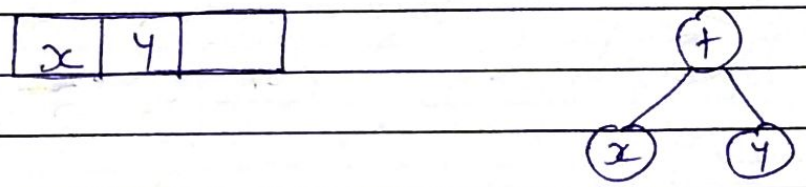
→ 1, 17, 29, 11, 52, 59, 53, 73, 98, 67, 34

(8)

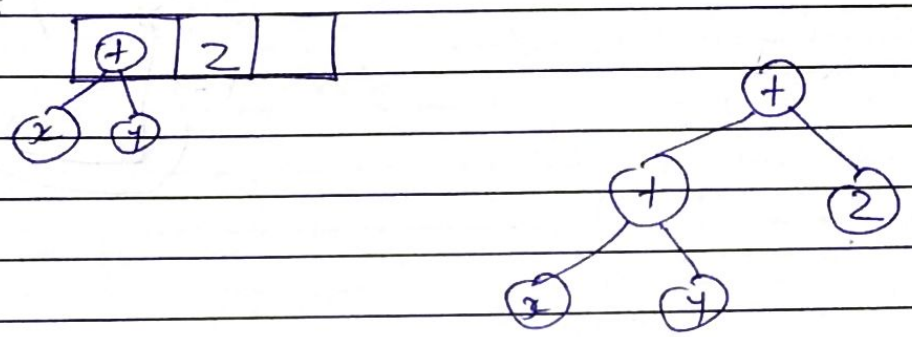
Q.3 Create Expression Tree from the equation in Q.1

1) Expression : $xy + 2 + ab + c - *$

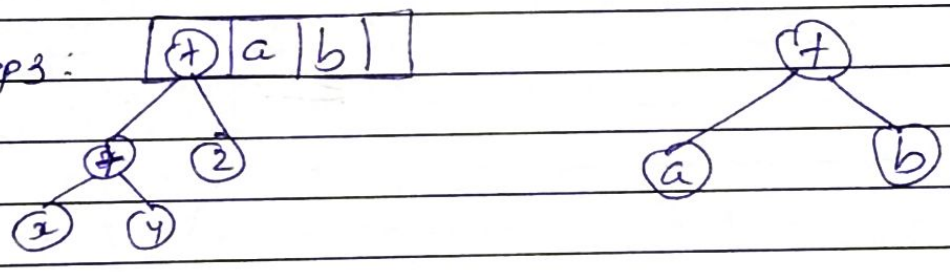
Step 1 :



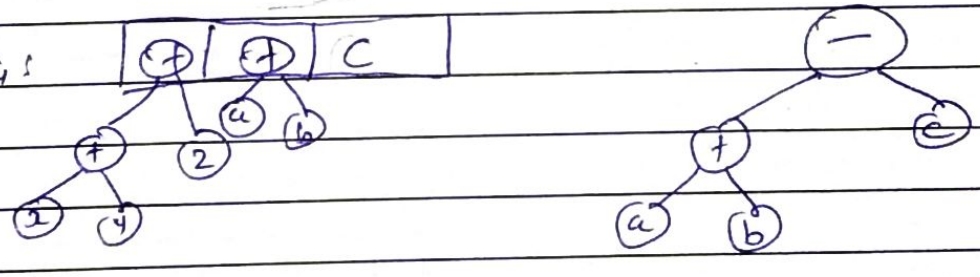
Step 2 :



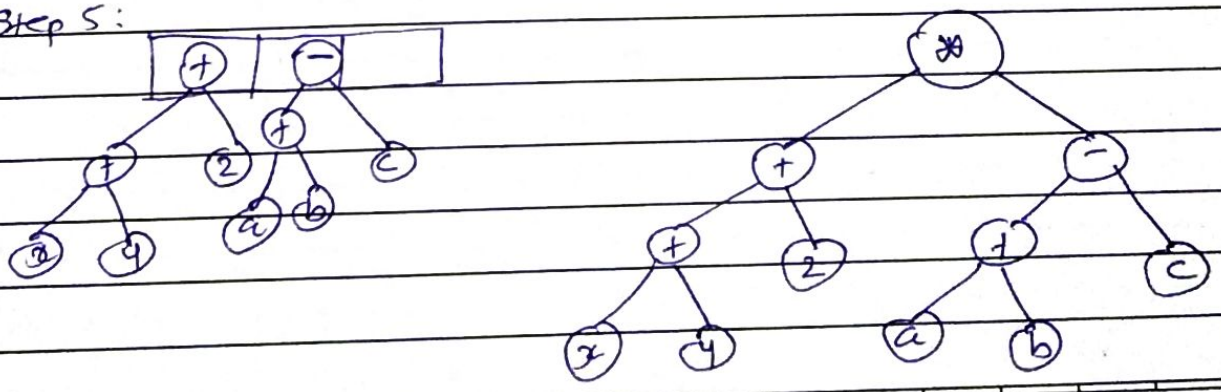
Step 3 :



Step. 4 :



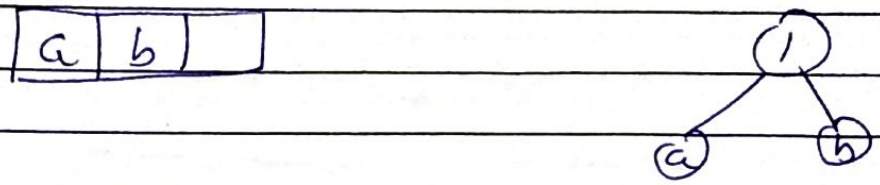
Step 5 :



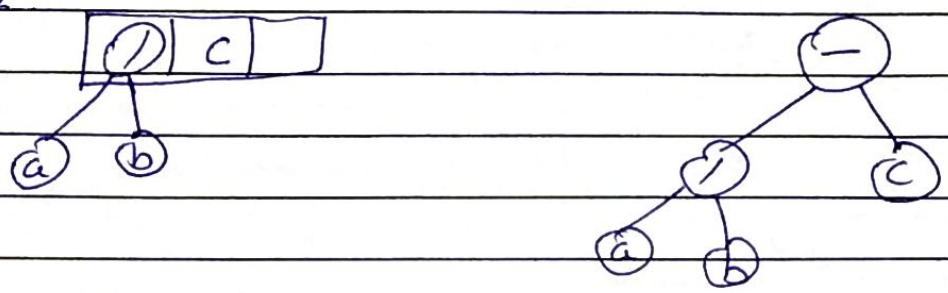
(9)

2) Expression: ab/c - de * +

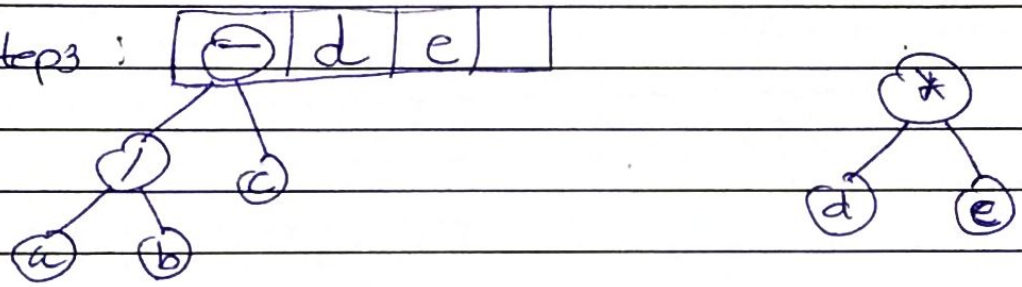
Step 1:



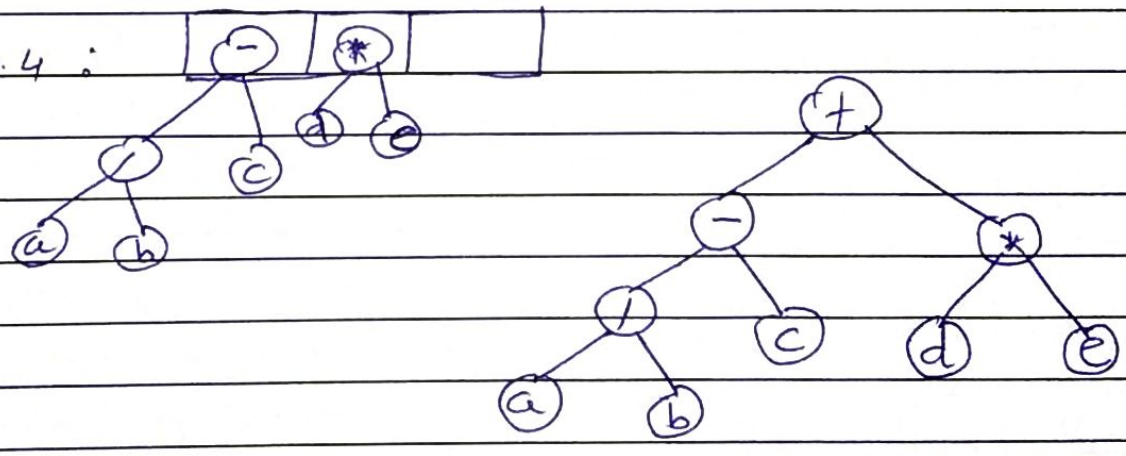
Step 2:



Step3 :

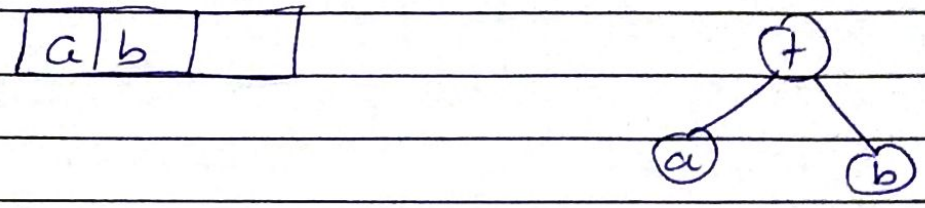


Step. 4 :

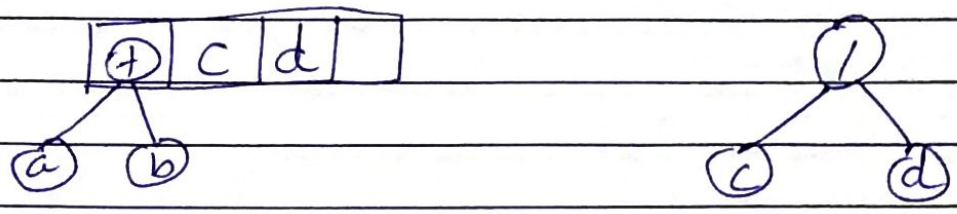


3) Expression : ab + cd / + efg * - *

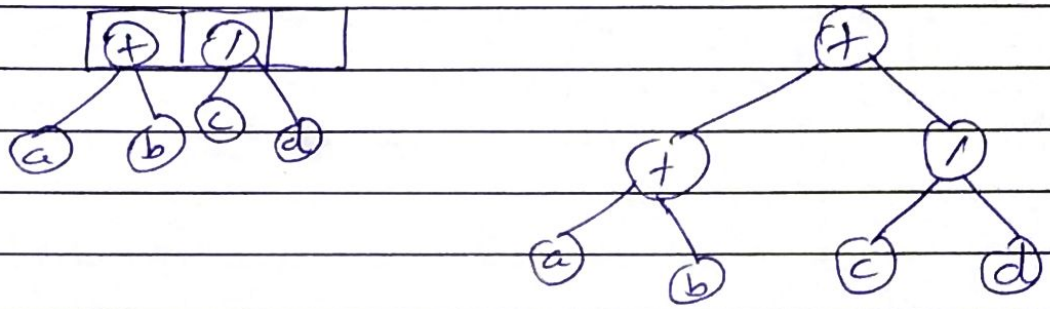
Step 1 :



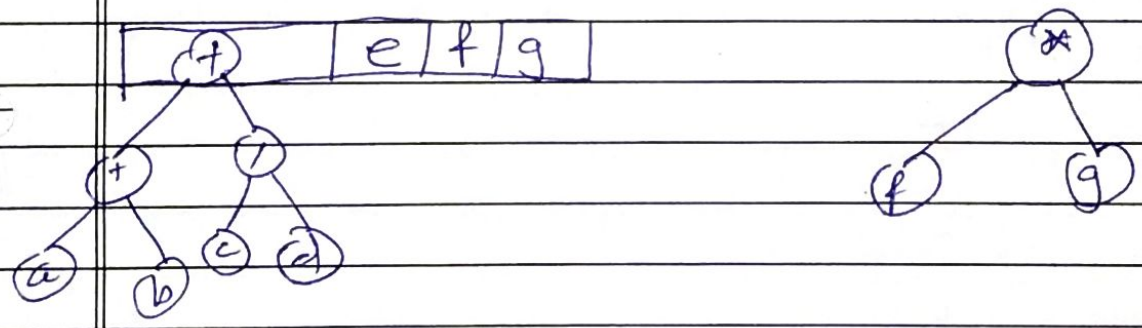
Step 2 :



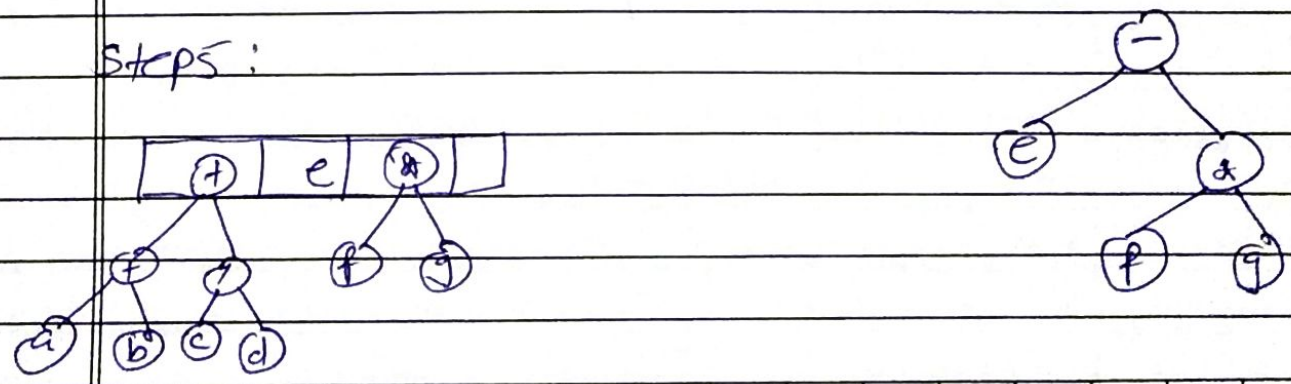
Step 3 :



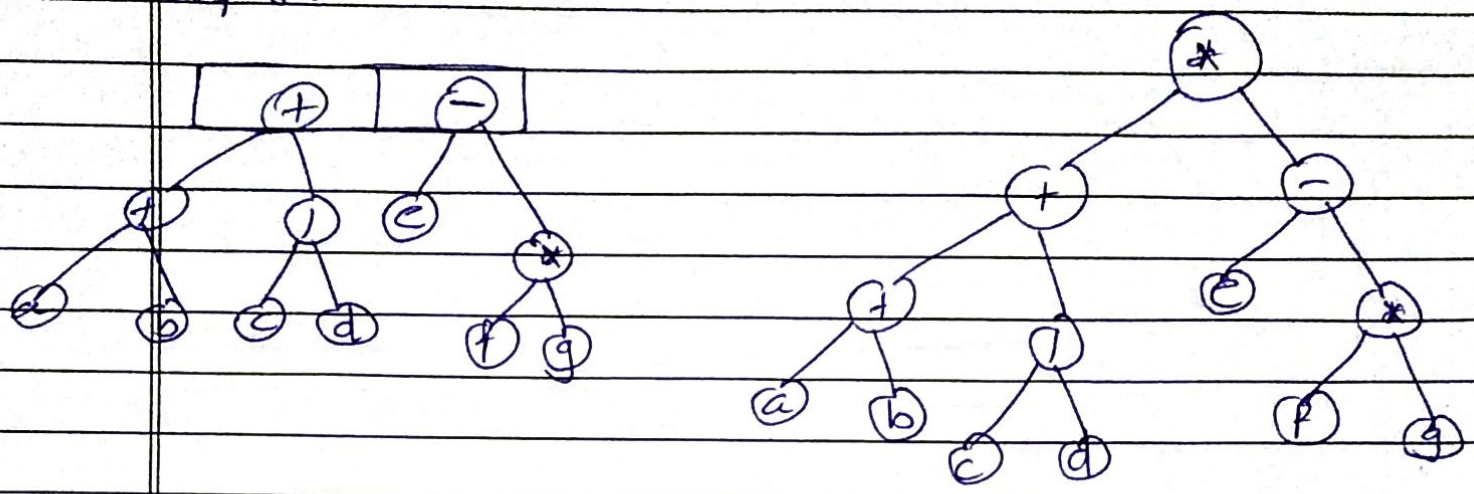
Step 4 :



Step 5 :

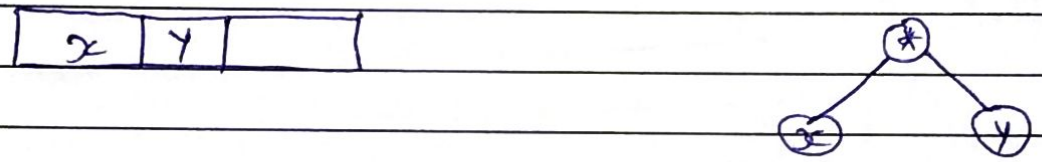


Step 6:

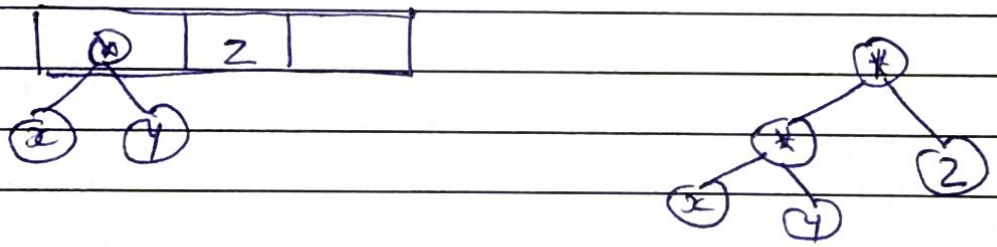


4) Expression: xy * z * w * ABC * + D - -

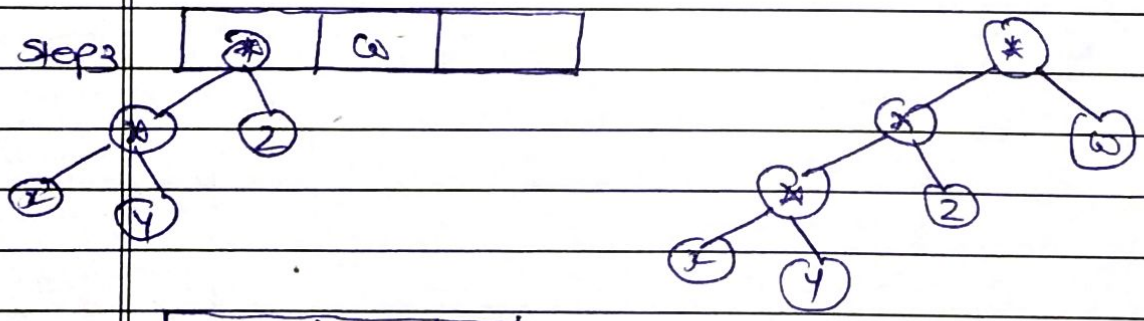
Step 1



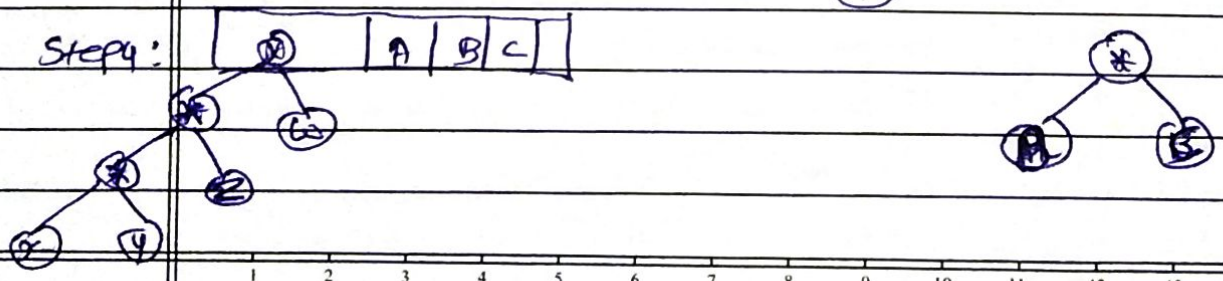
Step 2



Step 3

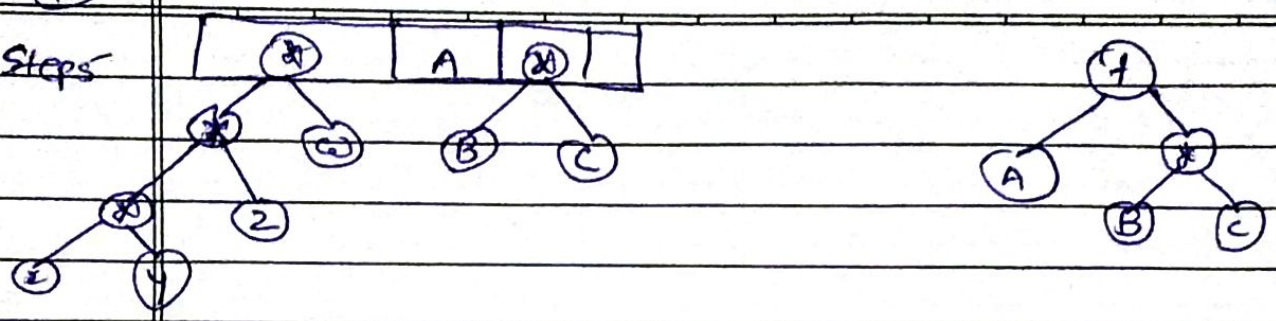


Step 4:

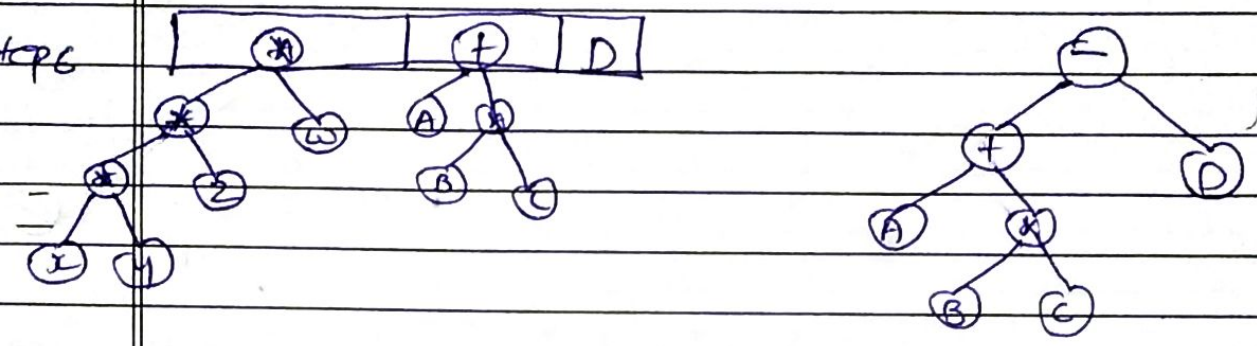


(12)

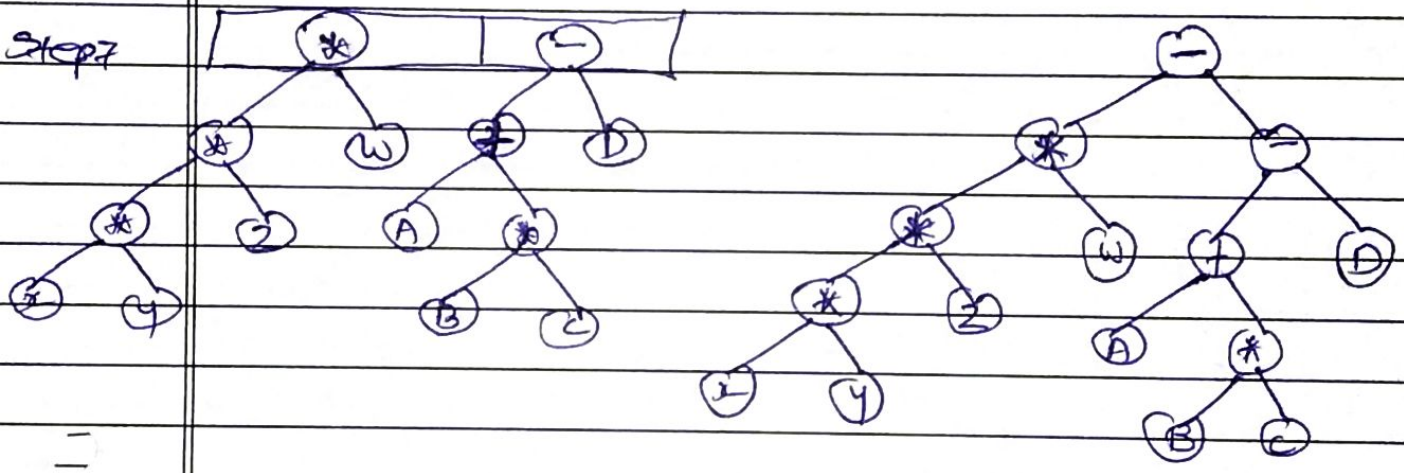
Step5



Step6

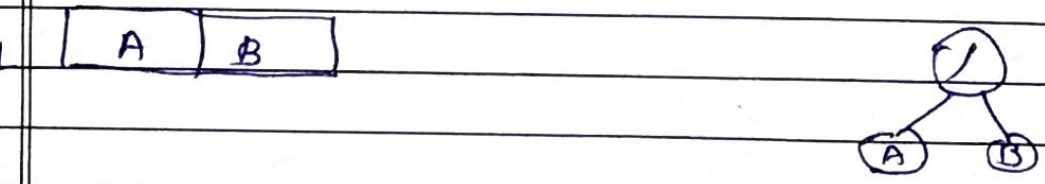


Step7

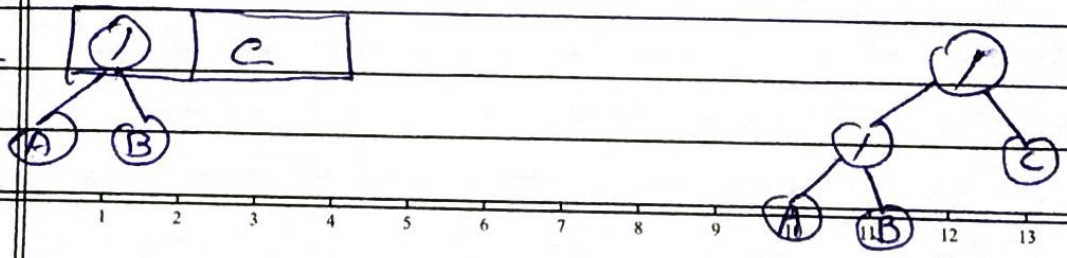


(5) Expression: AB/C/DE*+F-

Step1

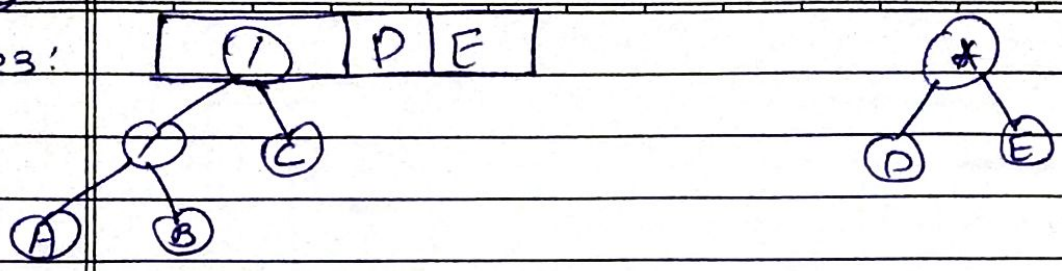


Step2

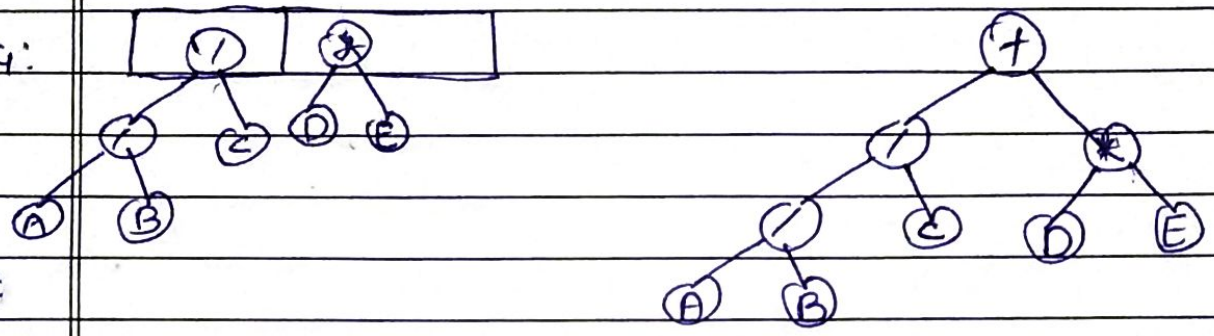


(13)

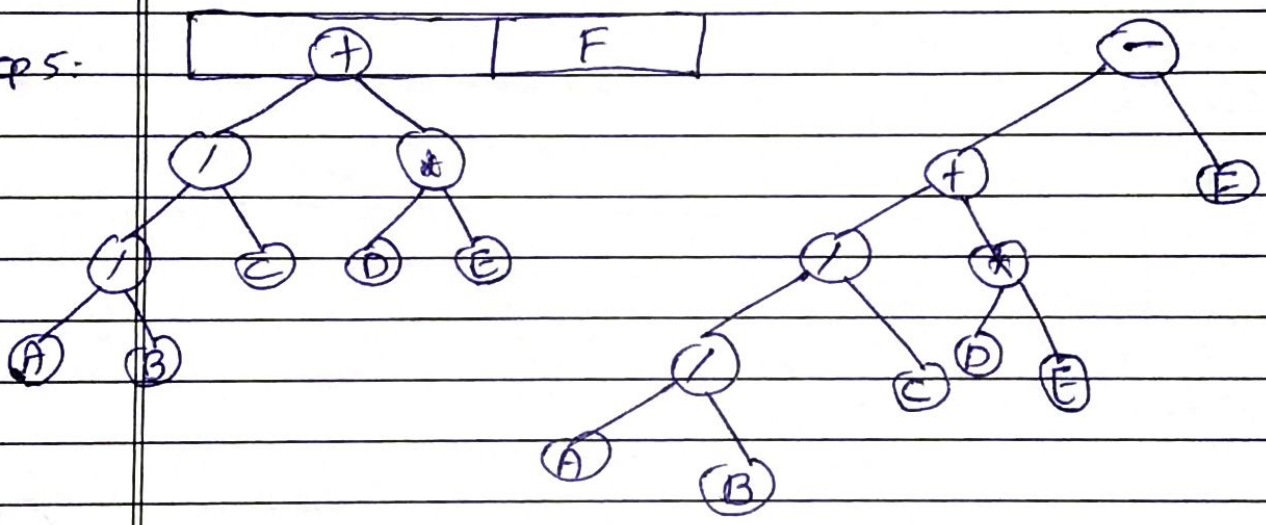
Step 3:



Step 4:

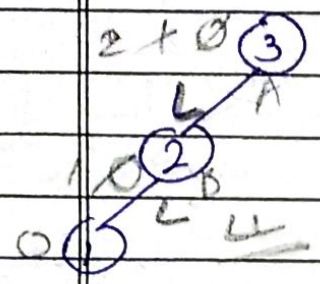


Step 5:

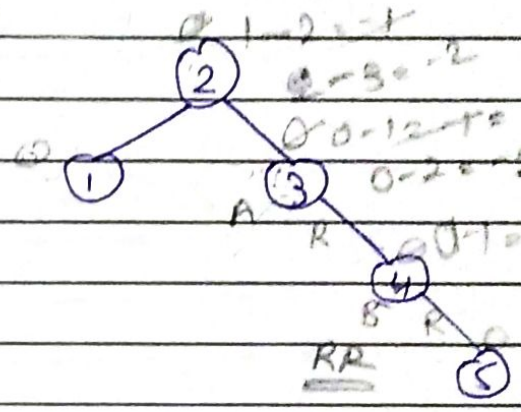


Q.4 AVL Tree (Rotation: LL, RR, LR, RL) (Balance factor: $h^L - h^R$ (-1, 0, 1))

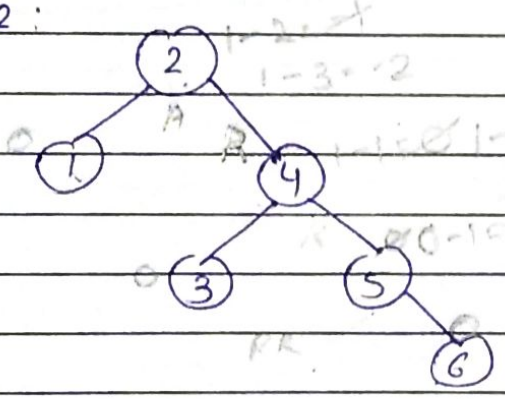
1) 3, 2, 1, 4, 5, 6, 7, 16, 15, 14, 13, 12, 11, 10, 8, 9



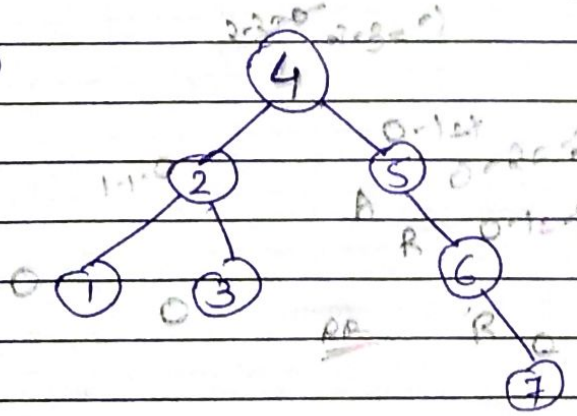
Rotation 1:



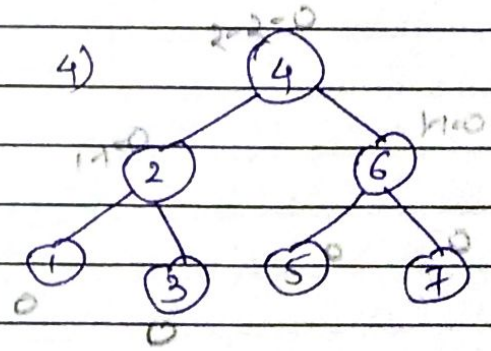
Rotation 2:



3.)

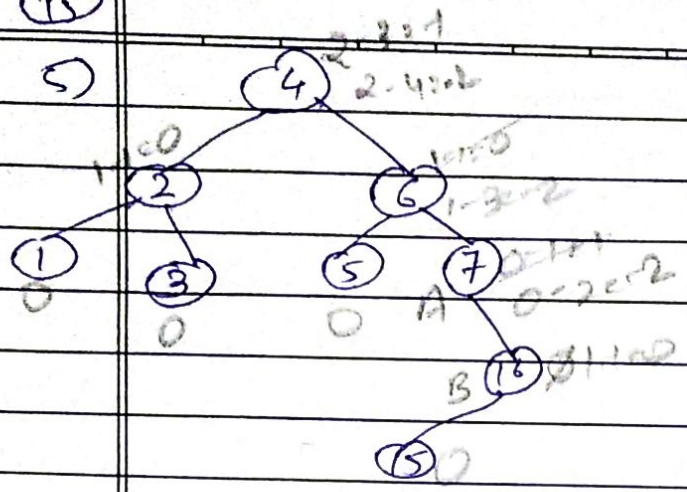


4)

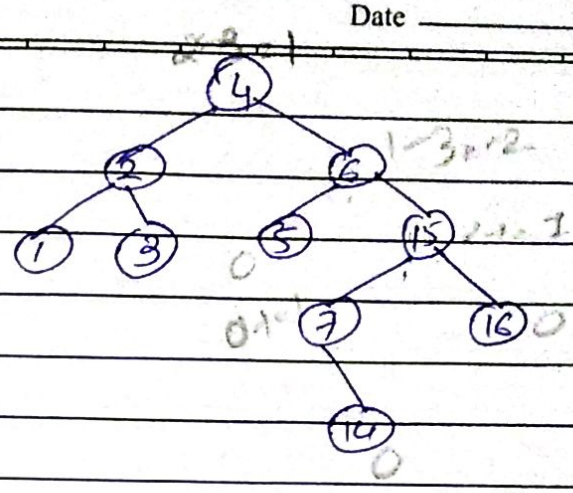


(15)

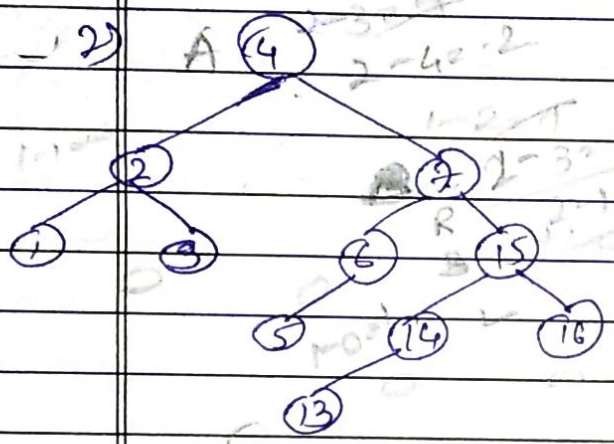
5)



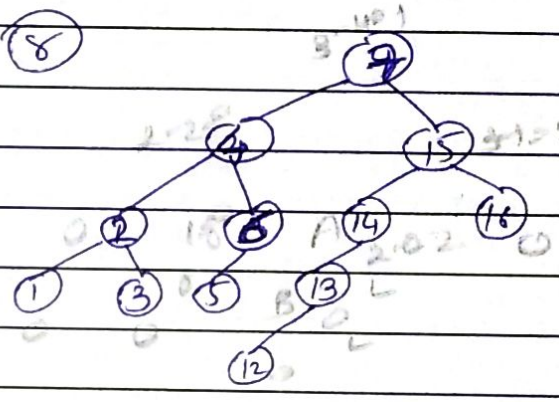
6)



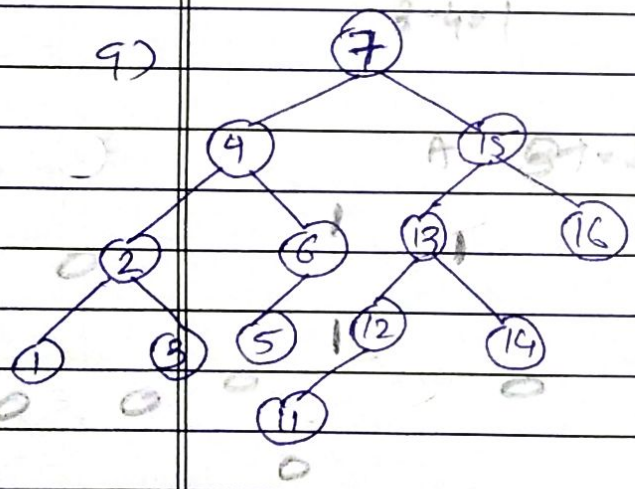
7)



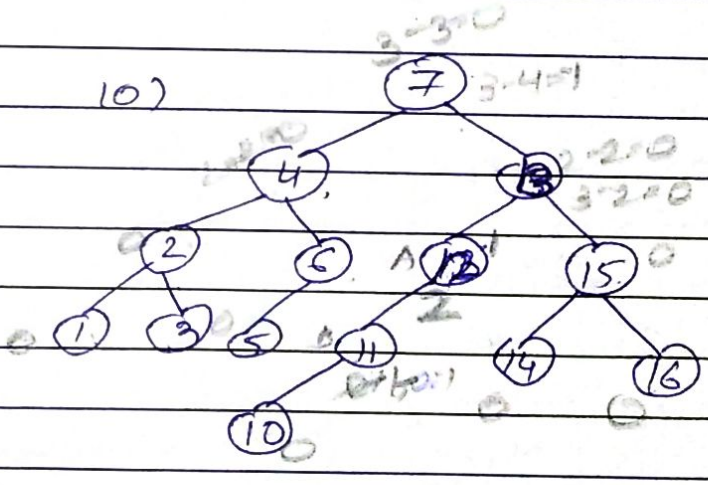
8)



9)

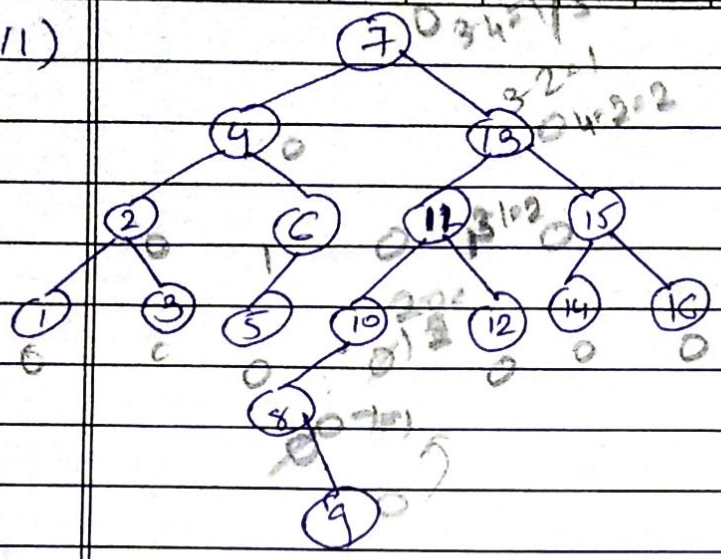


10)

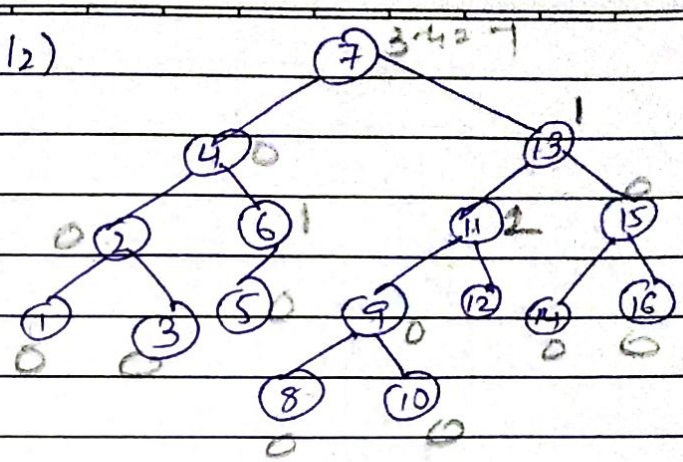


(16)

1)

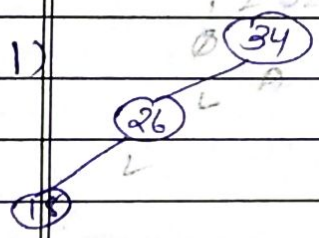


2)

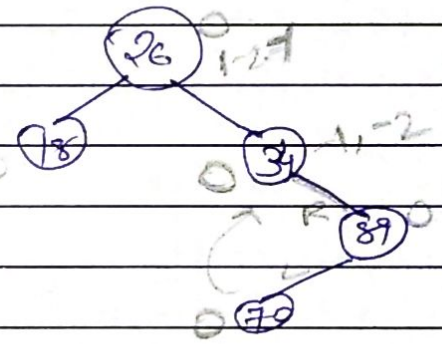


2) 34, 26, 18, 89, 70, 91, 2, 4, 6, 17, 48, 55, 10

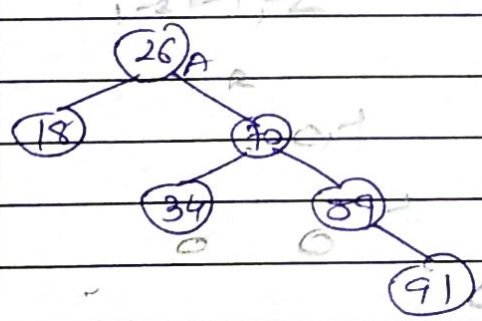
1)



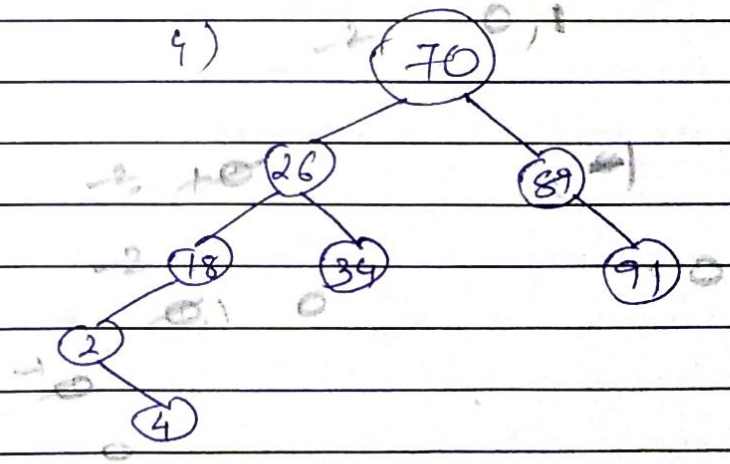
(P)



(3)

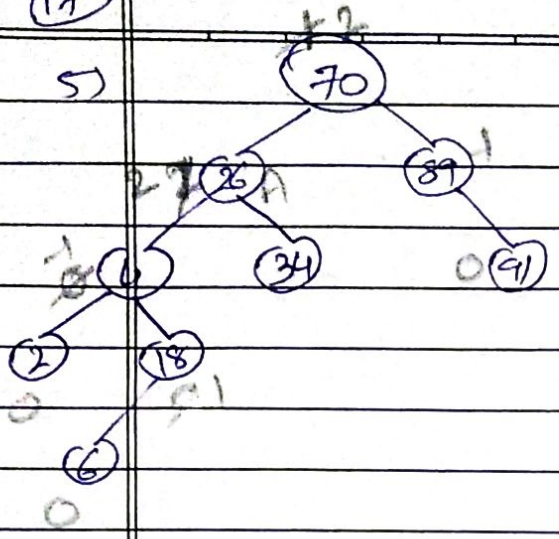


4)

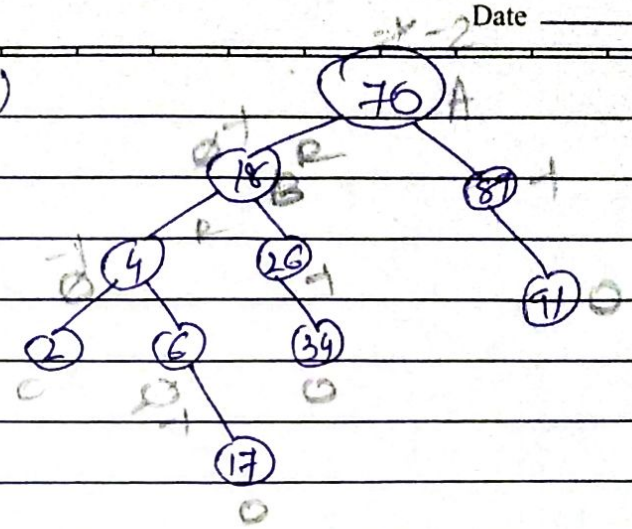


(17)

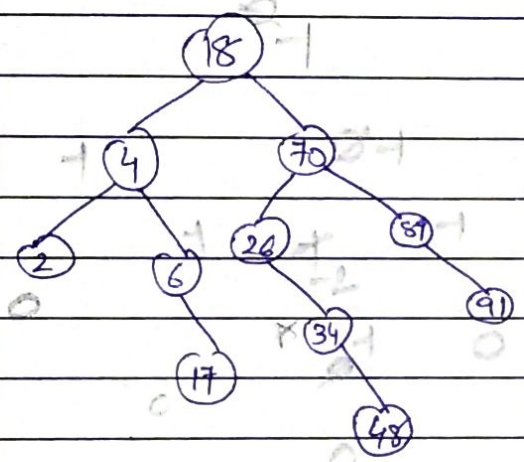
5)


70
26
89
6
34
91
2
18
6

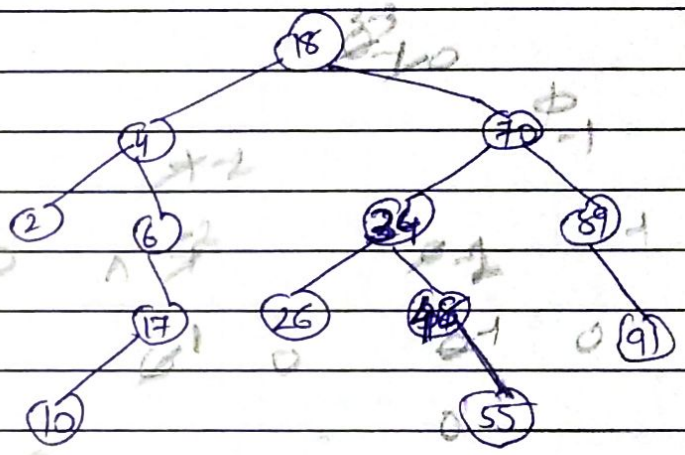
6)


70
18
89
4
26
91
2
6
34
17

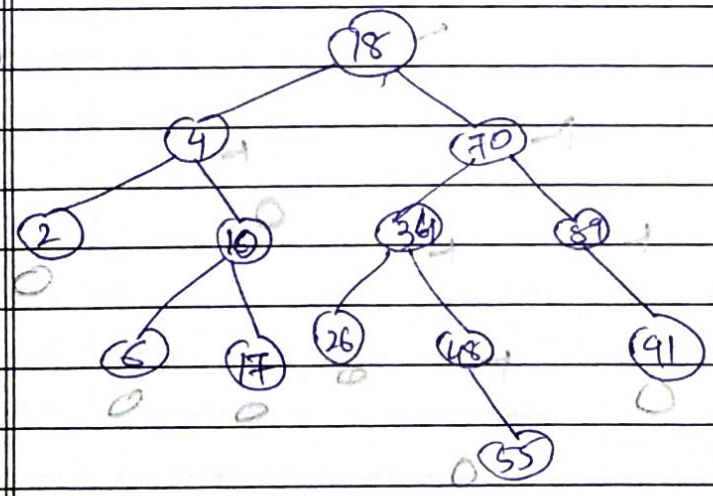
7)


18
4
70
2
6
26
89
34
91
17
48

8)


18
4
70
2
6
34
89
17
26
91
10
55

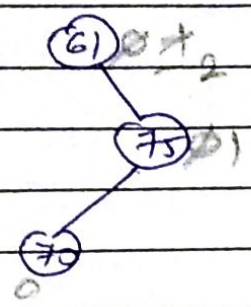
9)


18
4
70
2
10
34
89
6
17
26
48
91
55

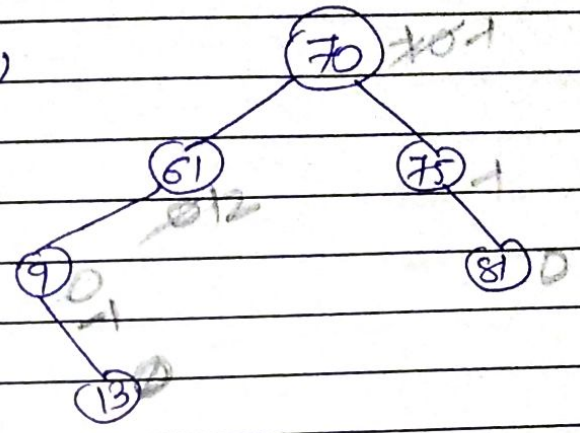
(18)

3) 61, 75, 70, 81, 9, 13, 46, 51, 11, 7, 88, 99, 12, 15.

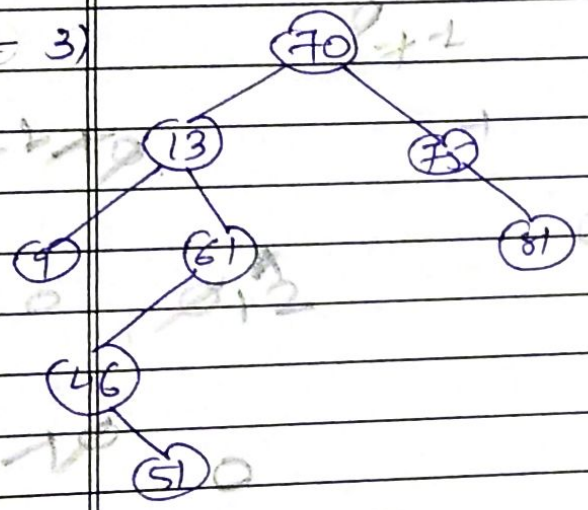
1)



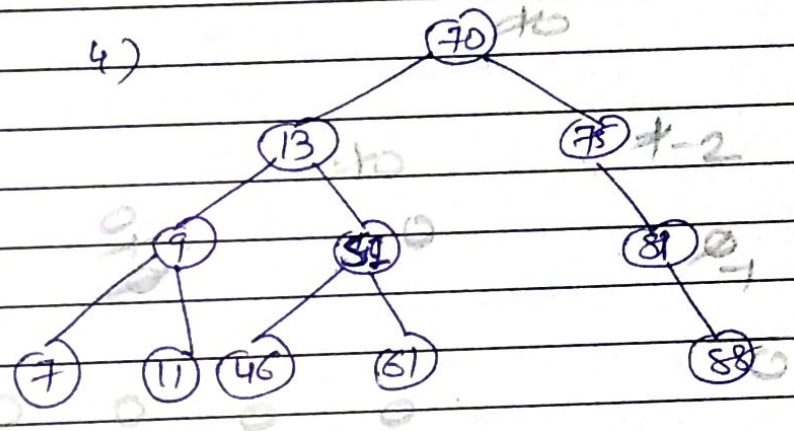
2)



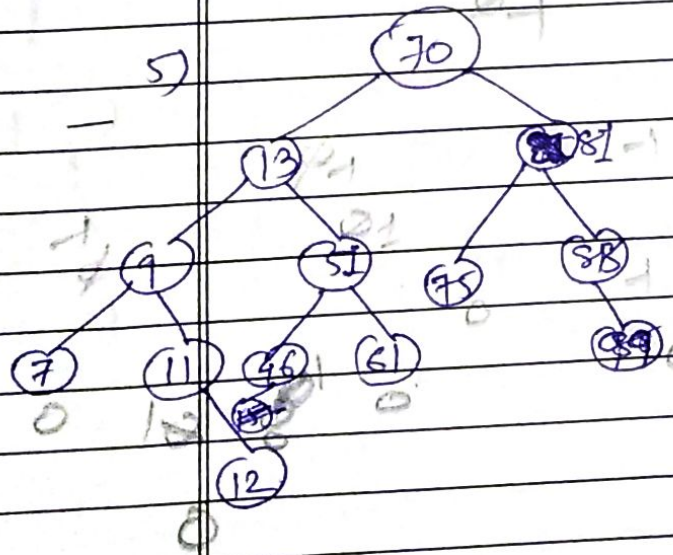
3)



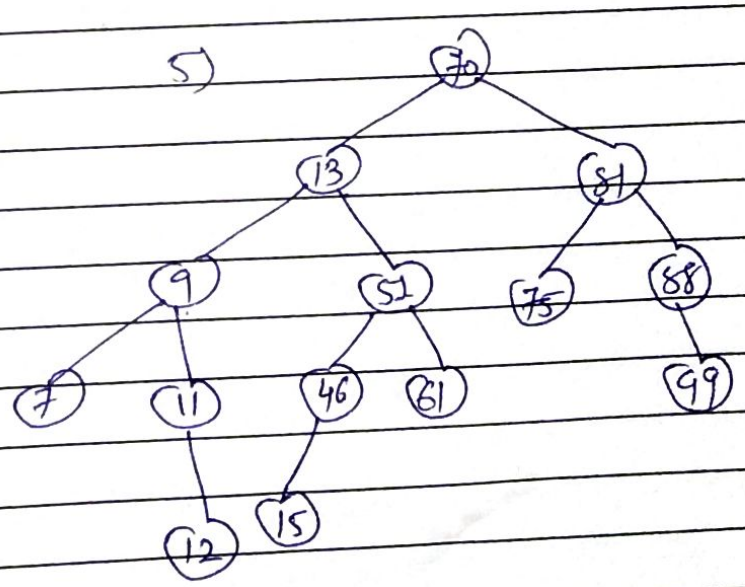
4)



5)



5)

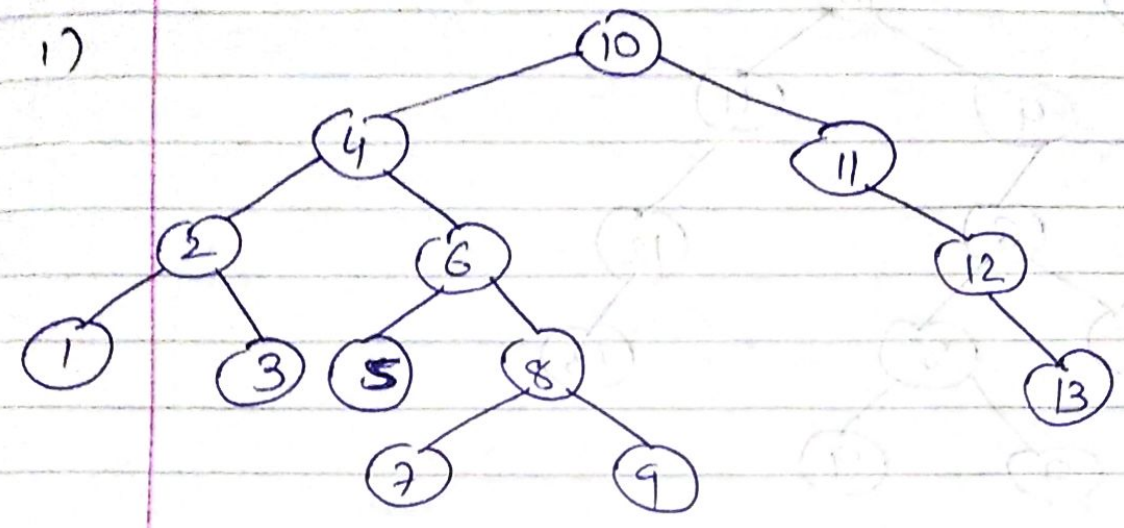


1

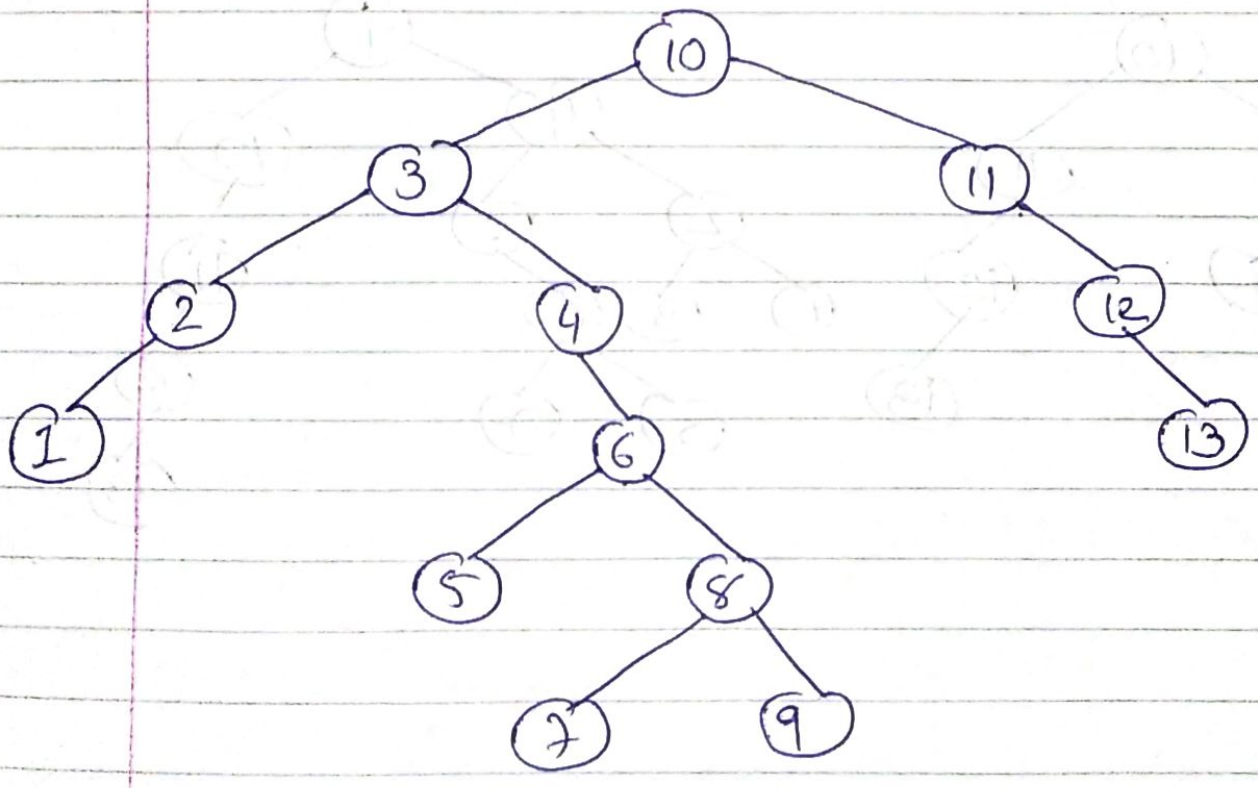
Page
Date

Splay Tree:

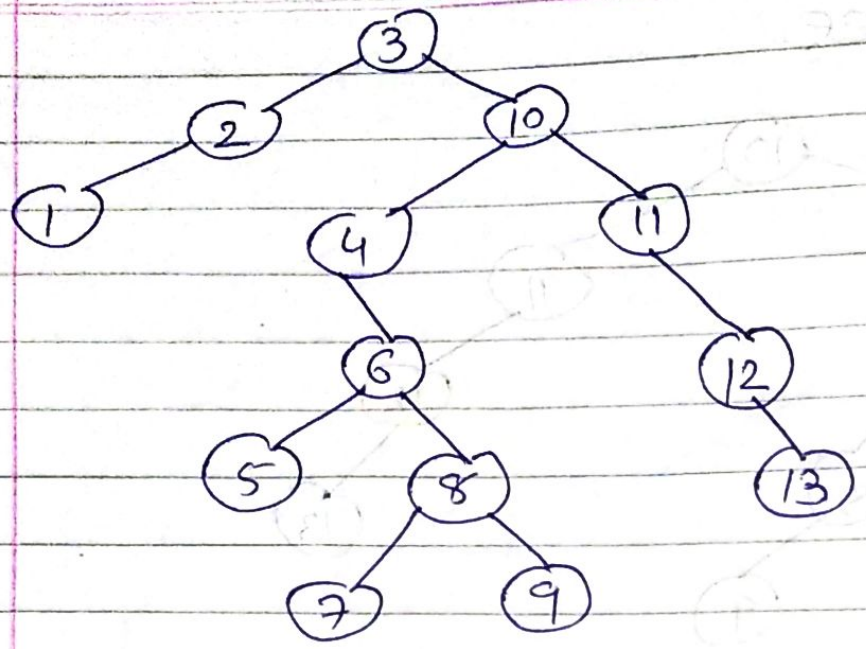
1)



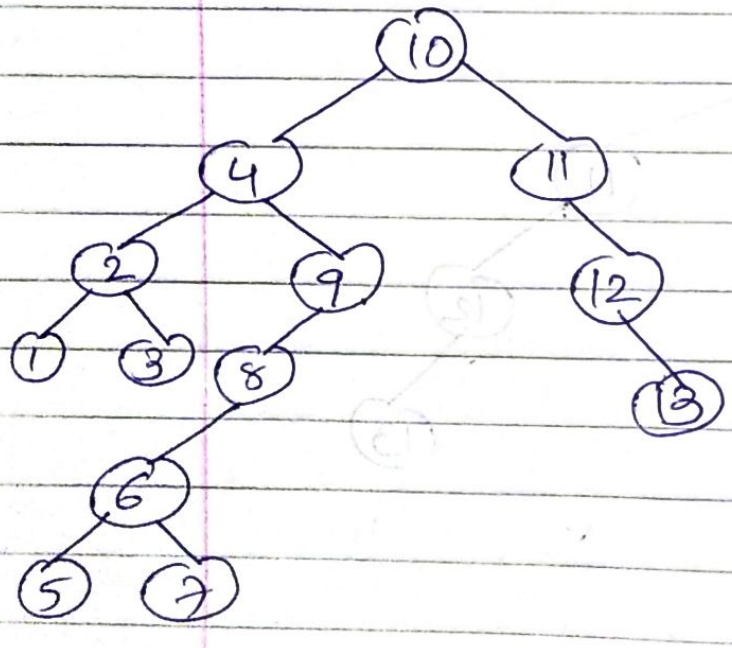
splay 3:

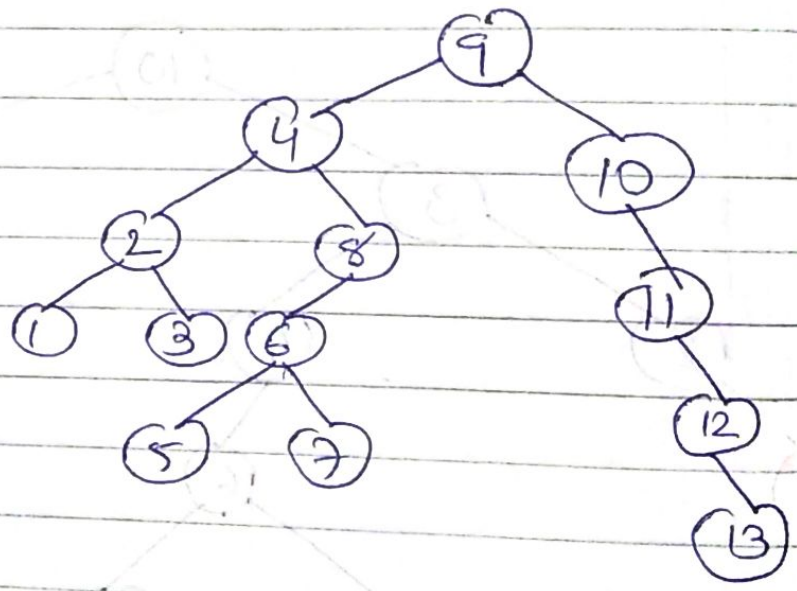


(2)


3
2
10
1
4
11
6
12
5
8
13
7
9

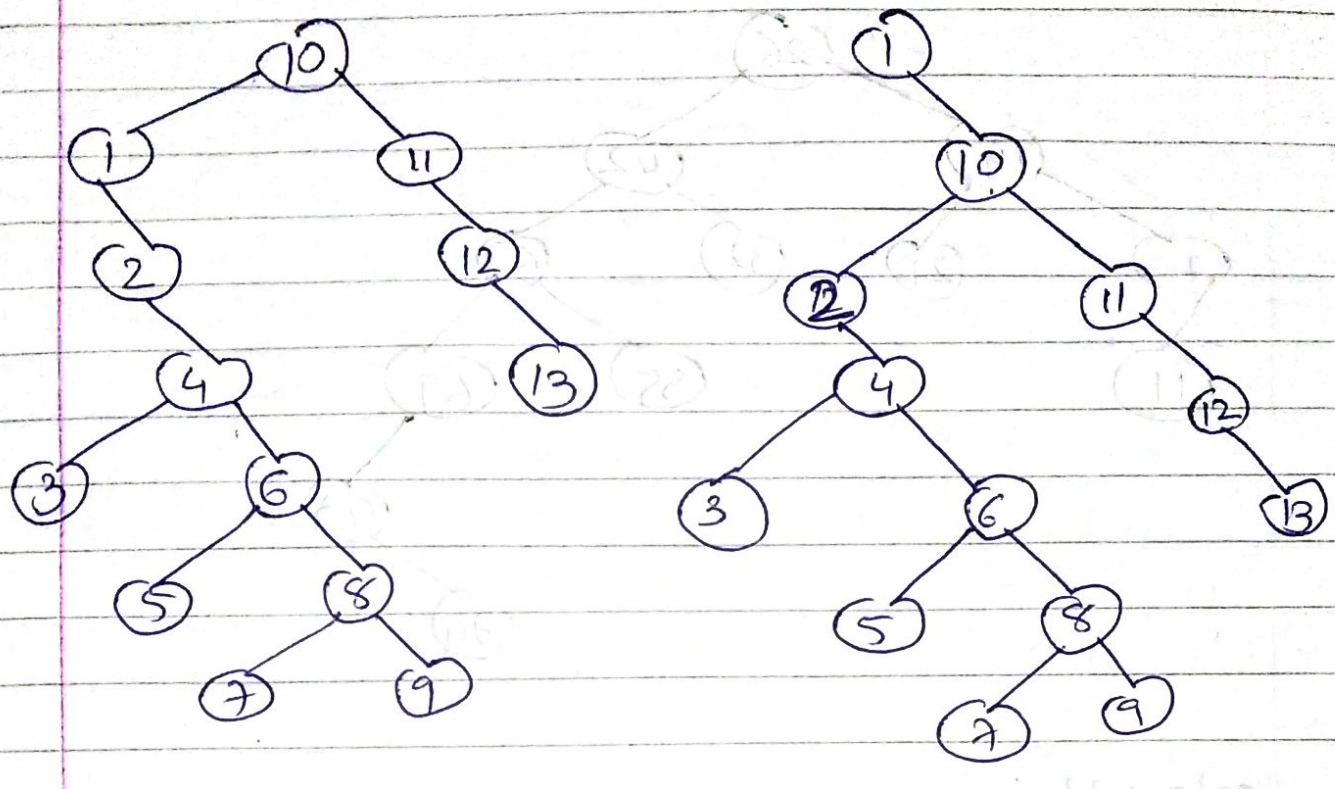
Splay 9:


10
4
11
2
9
12
1
3
8
13
6
5
7

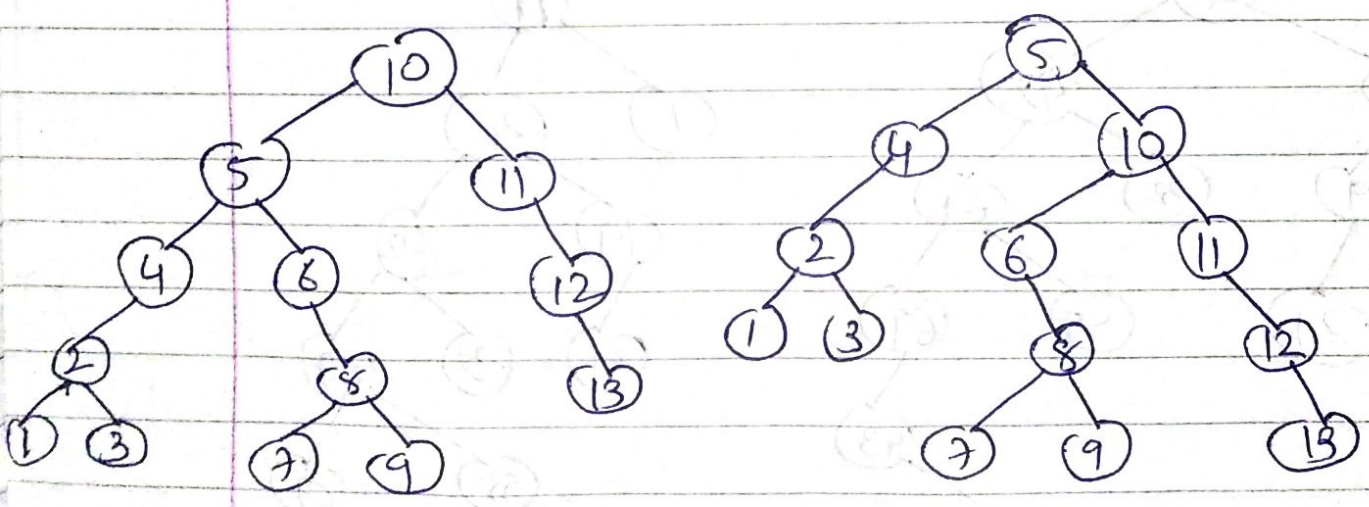

9
4
10
2
8
11
1
3
6
12
5
7
13

③

Splay 1:

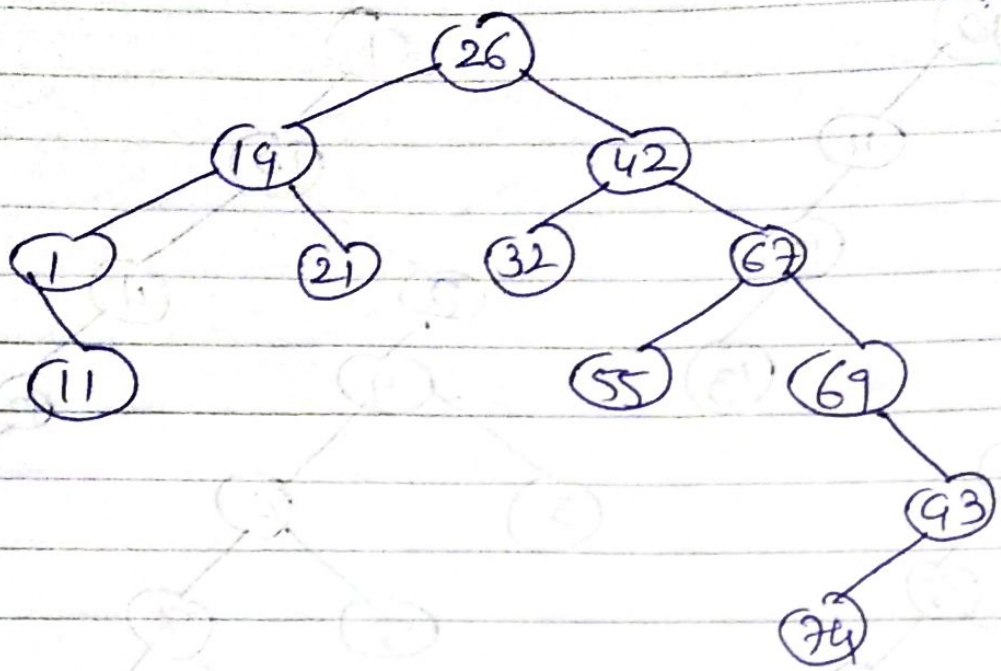


Splay 5:

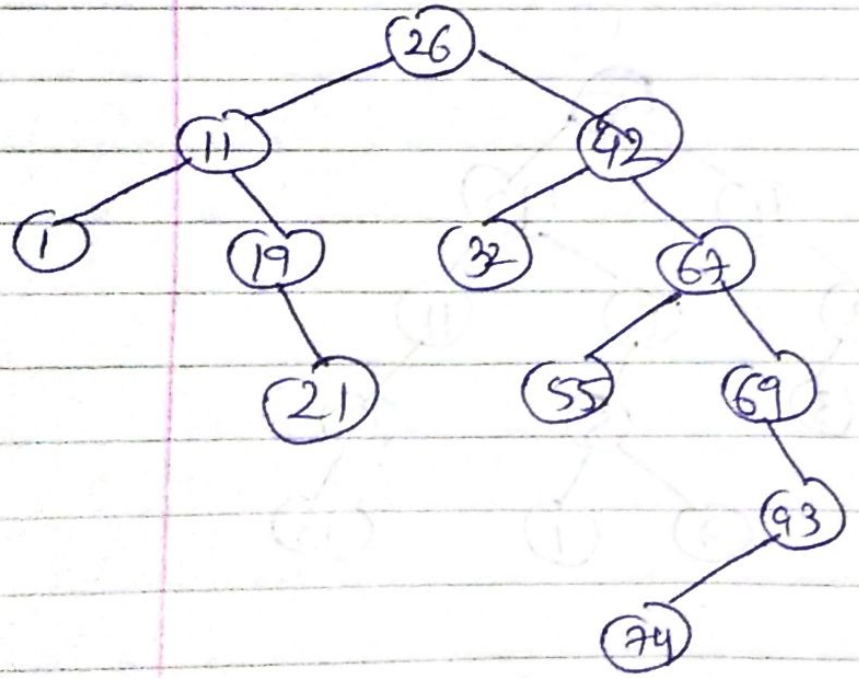


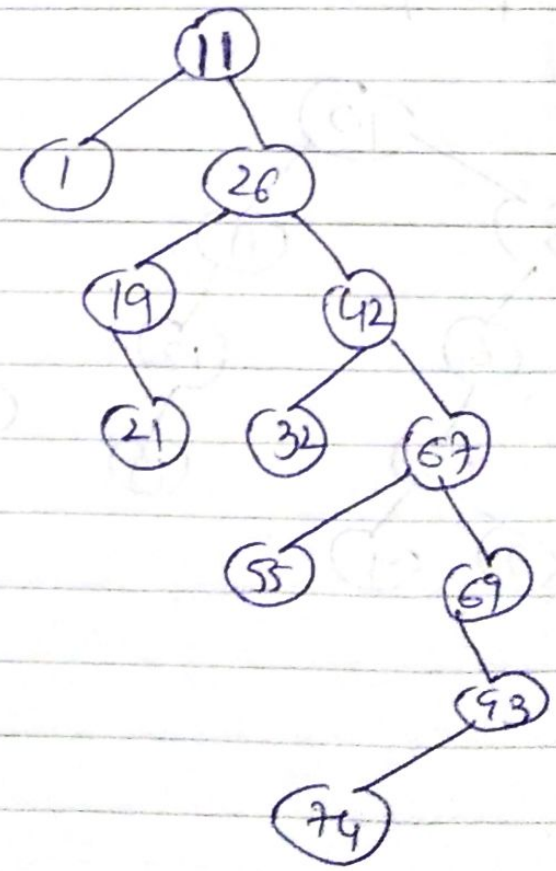
(4)

2) 26, 19, 42, 67, 21, 1, 69, 55, 93, 74, 32, 11



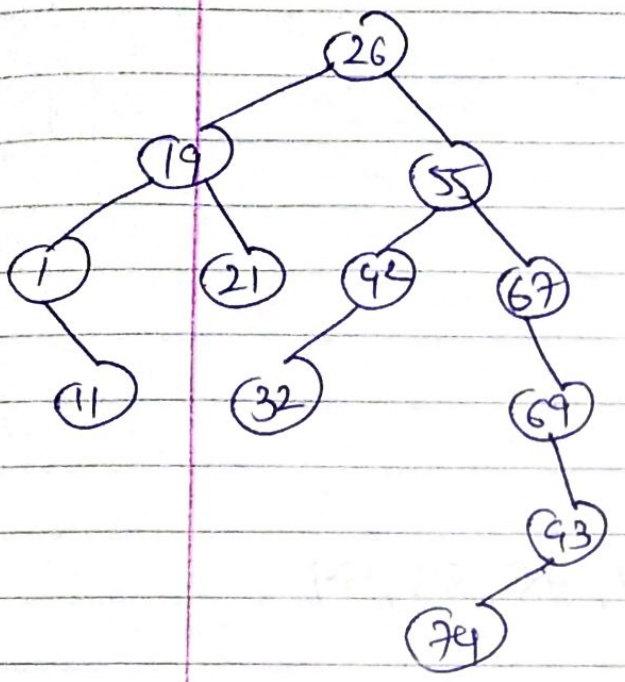
splay 11 :

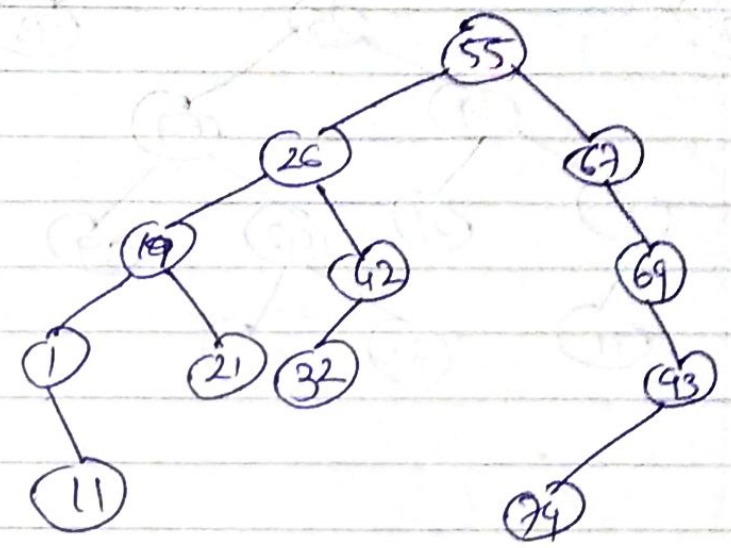




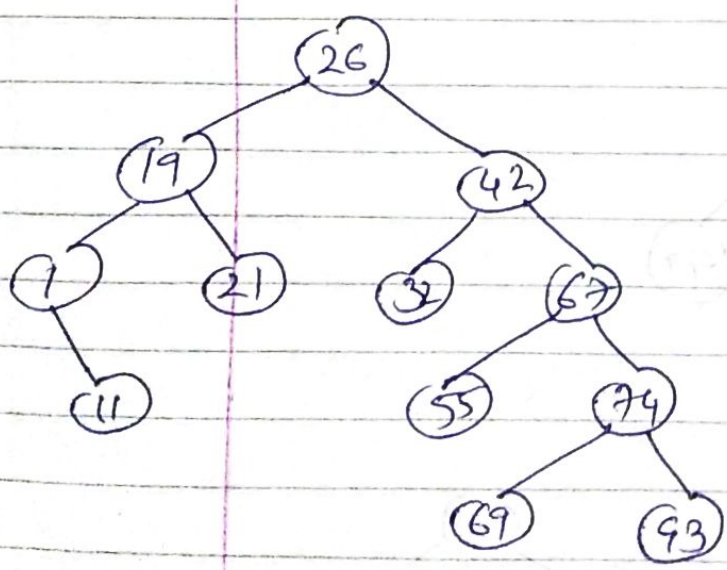
Page
Date

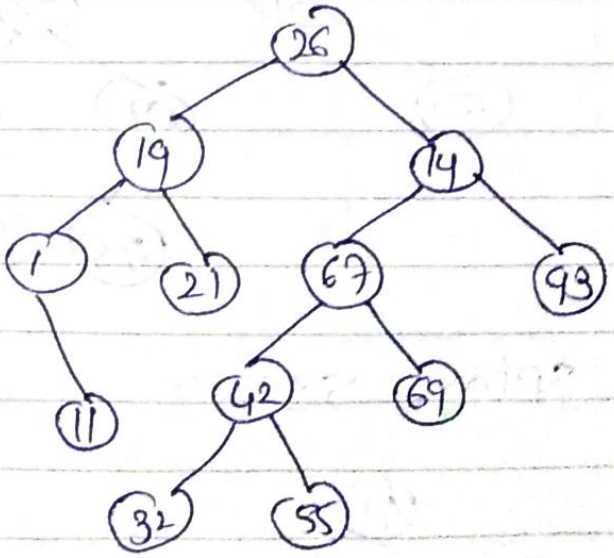
Splay 55



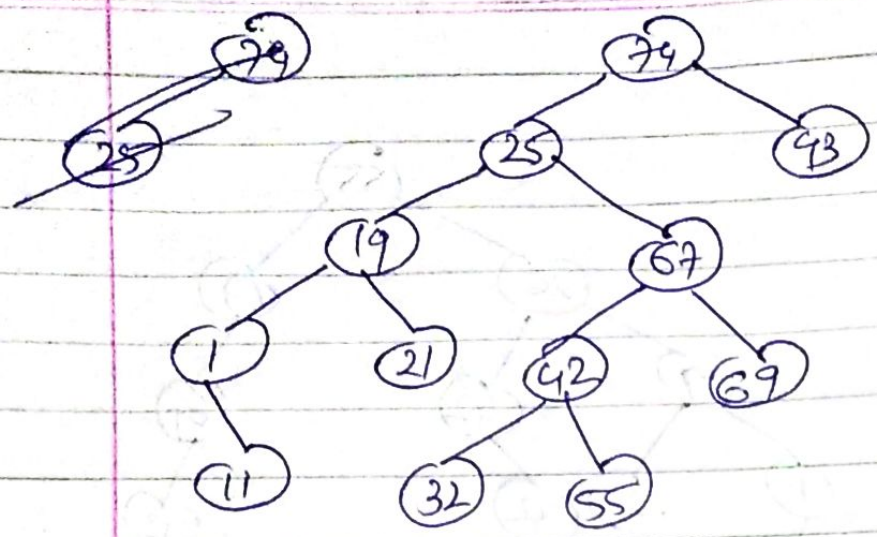


Splay 74:

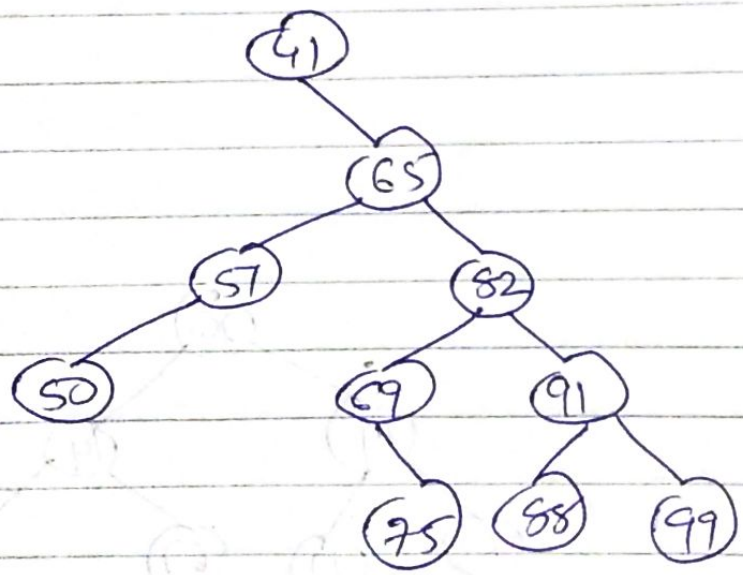




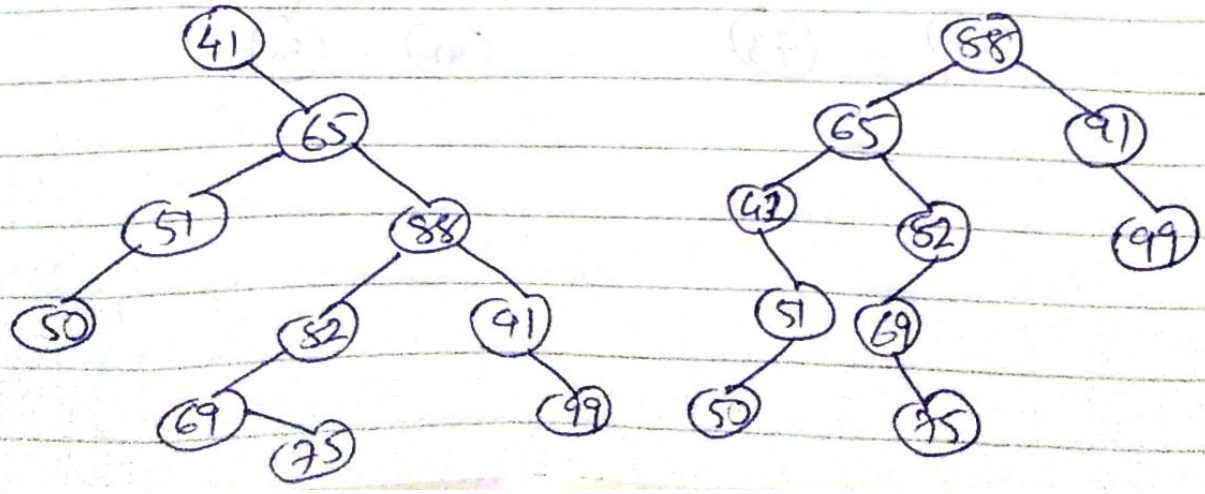
6



3) 41, 65, 82, 91, 51, 50, 69, 75, 88, 99



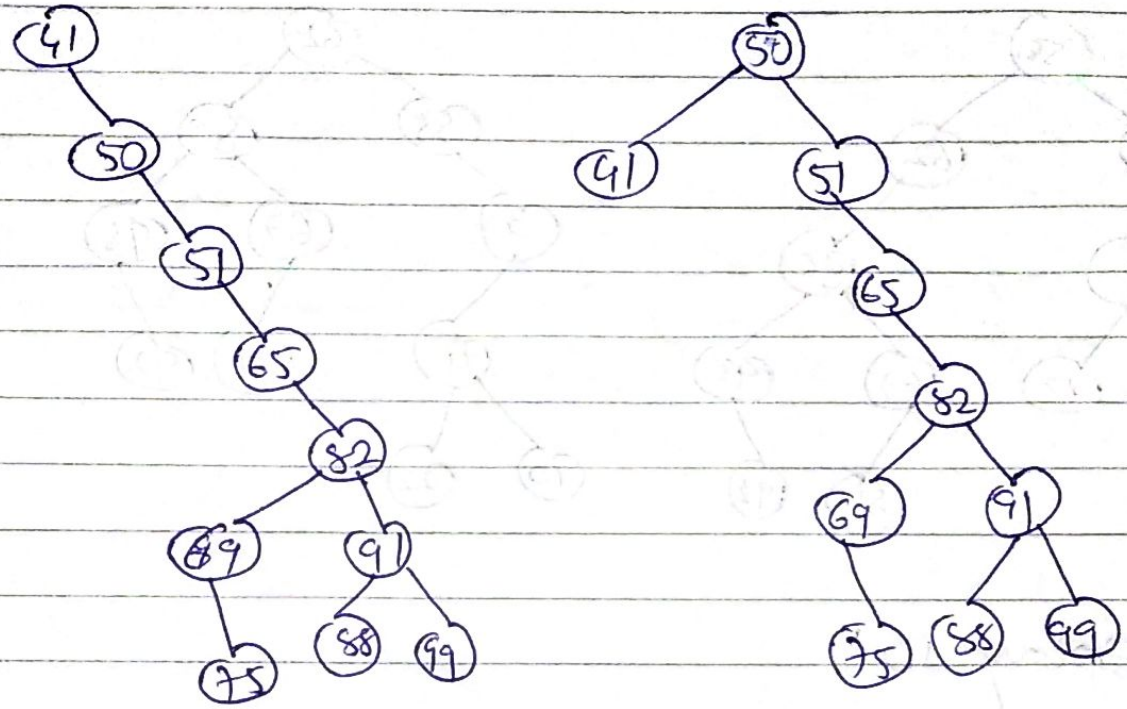
Splay 88.



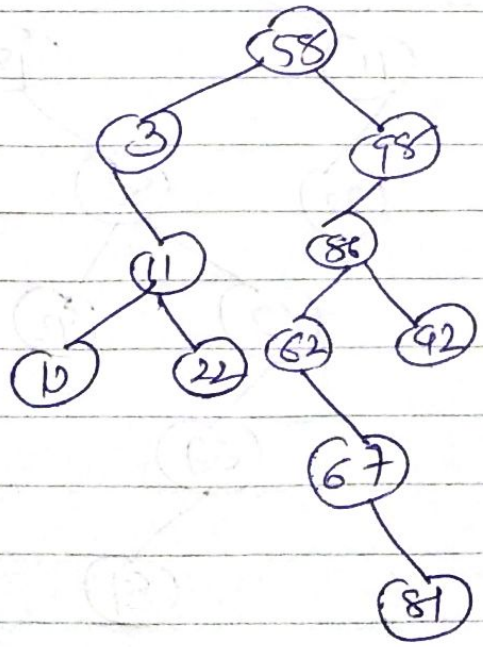
7

Page
Date

Splay 50:



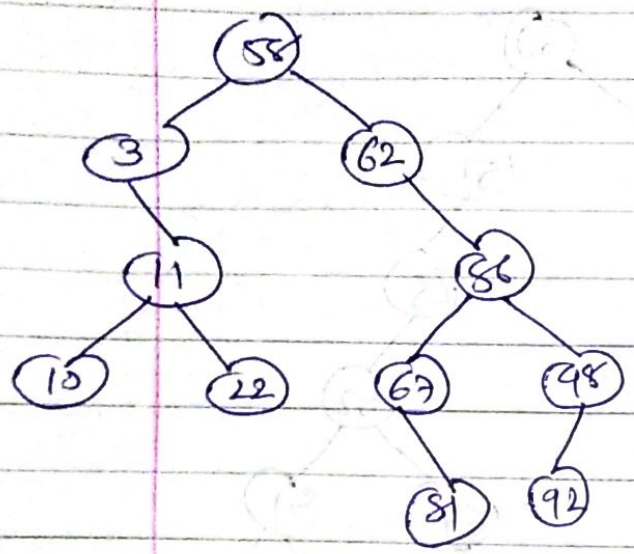
8) 58, 3, 98, 11, 86, 22, 62, 67, 81, 92, 10.

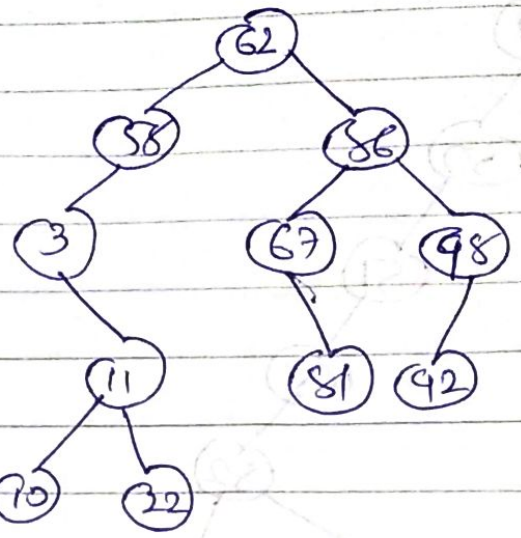


8.

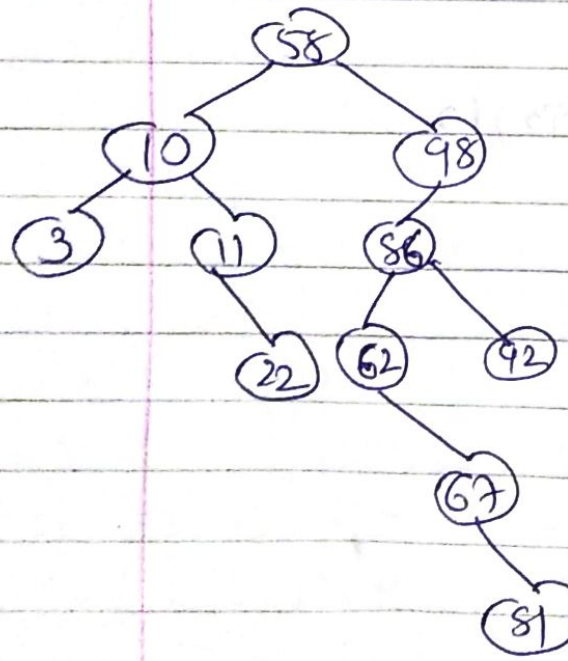
Page
Date

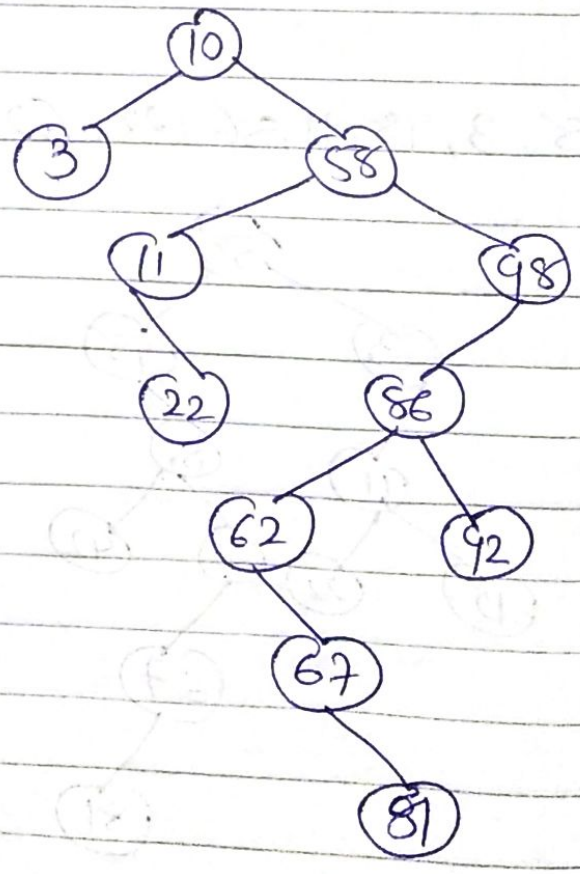
Splay 62:





Splay 10:

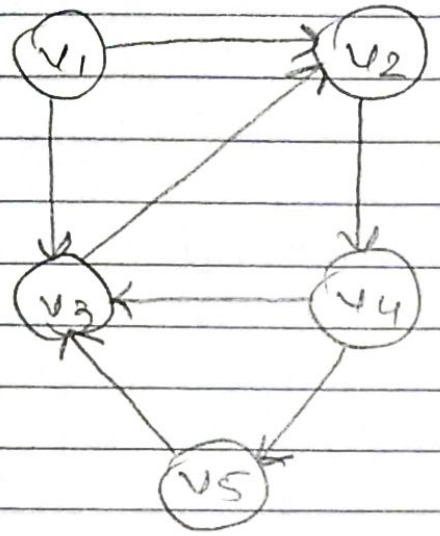




Assignment 2

Q.1 Unweighted Shortest Path: (Current Node + 1)

1)



v_1 is starting Node.

1) Initial state:

	Known	d_v	P_v
v_1	0	∞	0
v_2	0	∞	0
v_3	0	∞	0
v_4	0	∞	0
v_5	0	∞	0

2) v_1

	Known	d_v	P_v
v_1	1	0	0
v_2	0	1	v_1
v_3	0	1	v_1
v_4	0	∞	0
v_5	0	∞	0

3) v_2

	Known	d_v	P_v
v_1	1	0	0
v_2	1	1	v_1
v_3	0	1	v_1
v_4	0	2	v_2
v_5	0	∞	0

4) v_3

	Known	d_v	P_v
v_1	1	0	0
v_2	1	1	v_1
v_3	1	1	v_1
v_4	0	2	v_2
v_5	0	∞	0

DELUXE
PAGE NO.:
DATE:

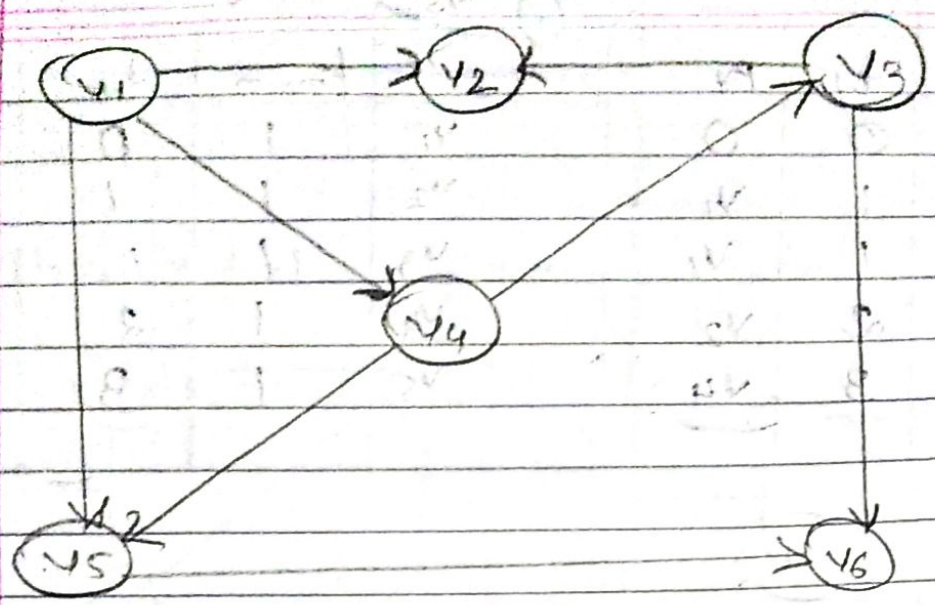
5) v_4

	Known	d_v	p_v
v_1	1	0	0
v_2	1	1	v_1
v_3	1	1	v_1
v_4	1	2	v_2
v_5	0	3	v_4

6) v_5

	Known	d_v	p_v
v_1	1	0	0
v_2	1	1	v_1
v_3	1	1	v_1
v_4	1	2	v_2
v_5	1	3	v_4

2)



V_1 is starting node;

1) Initial state

	Known	d_v	P_v
V_1	0	∞	0
V_2	0	∞	0
V_3	0	∞	0
V_4	0	∞	0
V_5	0	∞	0
V_6	0	∞	0

2) V_1

	Known	d_v	P_v
V_1	1	0	0
V_2	0	1	V_1
V_3	0	∞	0
V_4	0	1	V_1
V_5	0	1	V_1
V_6	0	∞	0

3) V_2

	Known	d_v	P_v
V_1	1	0	0
V_2	1	1	V_1
V_3	0	∞	0
V_4	0	1	V_1
V_5	0	1	V_1
V_6	0	∞	0

4) V_4

	Known	d_v	P_v
V_1	1	0	0
V_2	1	1	V_1
V_3	0	2	V_4
V_4	1	1	V_1
V_5	0	1	V_1
V_6	0	∞	0

5) v_5

	known	dv	pv
v_1	1	0	0
v_2	1	1	v_1
v_3	0	2	v_4
v_4	1	1	v_1
v_5	1	1	v_1
v_6	0	~~3~~ 2	v_5

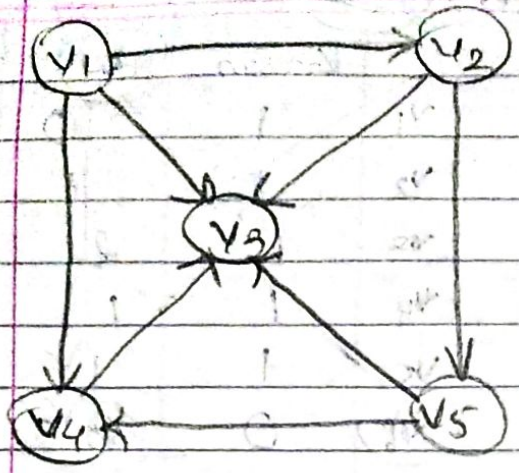
6) v_3

	known	dv	pv
v_1	1	0	0
v_2	1	1	v_1
v_3	1	2	v_4
v_4	1	1	v_1
v_5	1	1	v_1
v_6	0	~~3~~ 2	v_5

7) v_6

	known	dv	pv
v_1	1	0	0
v_2	1	1	v_1
v_3	1	2	v_4
v_4	1	1	v_1
v_5	1	1	v_1
v_6	1	~~3~~ 2	v_5

3)



→ V1 is Starting Node

Initial state

	Known	dv	Pv
V1	0	∞	0
V2	0	∞	0
V3	0	∞	0
V4	0	∞	0
V5	0	∞	0

2) V1

	Known	dv	Pv
V1	1	0	0
V2	0	1	V1
V3	0	1	V1
V4	0	1	V1
V5	0	∞	0

3) V2

	Known	dv	Pv
V1	1	0	0
V2	1	1	V1
V3	0	1	V1
V4	0	1	V1
V5	0	2	V2

4) V3

	Known	dv	Pv
V1	1	0	0
V2	1	1	V1
V3	1	1	V1
V4	0	1	V1
V5	0	2	V2

5) V4

	Known	dv	Pv
V1	1	0	0
V2	1	1	V1
V3	1	1	V1
V4	1	1	V1
V5	0	2	V2

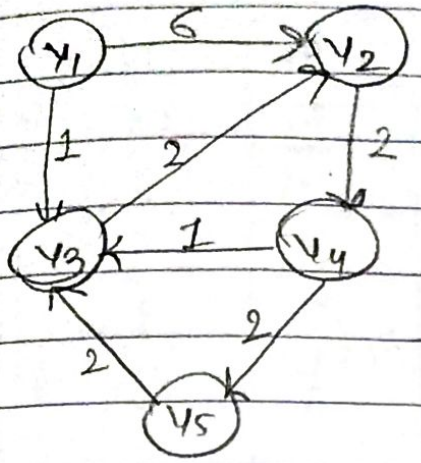
6) V5

	Known	dv	Pv
V1	1	0	0
V2	1	1	V1
V3	1	1	V1
V4	1	1	V1
V5	1	2	V2

PAGE NO.:
DATE:

Q2 Weighted Shortest Path.

1)



→ v_1 is Starting Node.

1) Initial state

	Known	dv	Pv
v_1	0	0	0
v_2	0	∞	0
v_3	0	∞	0
v_4	0	∞	0
v_5	0	∞	0

2) v_1

	Known	dv	Pv
v_1	1	0	0
v_2	0	6	v_1
v_3	0	1	v_1
v_4	0	∞	0
v_5	0	∞	0

3) v_3

	Known	dv	Pv
v_1	1	0	0
v_2	0	3	v_3
v_3	1	1	v_1
v_4	0	∞	0
v_5	0	∞	0

shortest path (6 replaced with 3)

4) v_2

	Known	dv	Pv
v_1	1	0	0
v_2	1	3	v_3
v_3	1	1	v_1
v_4	0	5 (3+2)	v_2
v_5	0	∞	0

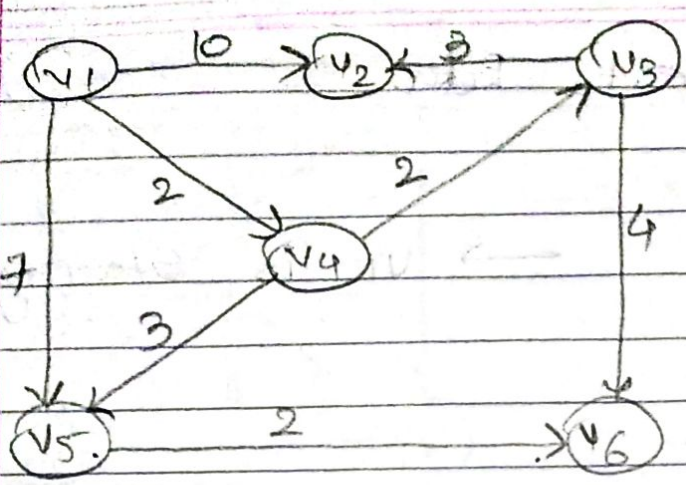
5) v_4

	Known	dv	Pv
v_1	1	0	0
v_2	1	3	v_3
v_3	1	1	v_1
v_4	1	5	v_2
v_5	0	7	v_4

6) v_5

	Known	dv	Pv
v_1	1	0	0
v_2	1	3	v_3
v_3	1	1	v_1
v_4	1	5	v_2
v_5	1	7	v_4

a)



→ v_1 is starting node.

1) Initial state.

	known	dv	pv
v_1	0	0	0
v_2	0	∞	0
v_3	0	∞	0
v_4	0	∞	0
v_5	0	∞	0
v_6	0	∞	0

2) v_1

	known	dv	pv
v_1	1	0	0
v_2	0	10	v_1
v_3	0	∞	0
v_4	0	2	v_1
v_5	0	7	v_1
v_6	0	∞	0

3) v_4

	known	dv	pv
v_1	1	0	0
v_2	0	10	v_1
v_3	0	4 (2+2)	v_4
v_4	1	2	v_1
7 to 5 (update) v_5	0	5 (2+3)	v_4
v_6	0	∞	0

4) v_3

	known	dv	pv
v_1	1	0	0
v_2	0	7 (4+3) (8+3)	v_3 (update)
v_3	1	4	v_4
v_4	1	2	v_1
v_5	0	5	v_4
v_6	0	8 (4+4)	v_3

DELUXE
PAGE NO.:
DATE:

5) V5

	known	dv	pv
V1	1	0	0
V2	0	7	V3
V3	1	4	V4
V4	1	2	V1
V5	1	5	V4
update V6	0	7 (5+2)	V5

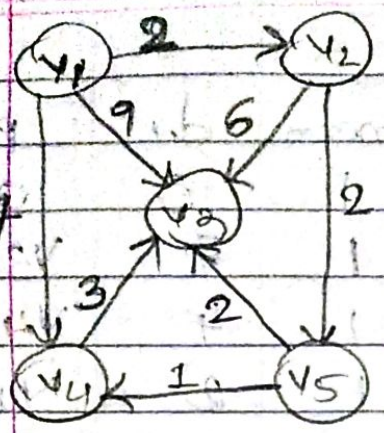
6) V2

	known	dv	pv
V1	1	0	0
V2	1	7	V3
V3	1	4	V4
V4	1	2	V1
V5	1	5	V4
V6	0	7	V5

7) V6

	known	dv	pv
V1	1	0	0
V2	1	7	V3
V3	1	4	V4
V4	1	2	V1
V5	1	5	V4
V6	1	7	V5

3)



→ v_1 is starting Node

1) Initial state

	known	d_v	P_v
v_1	0	0	0
v_2	0	∞	0
v_3	0	∞	0
v_4	0	∞	0
v_5	0	∞	0

2) v_1

	known	d_v	P_v
v_1	1	0	0
v_2	0	2	v_1
v_3	0	9	v_1
v_4	0	7	v_1
v_5	0	∞	0

3) v_2

	known	d_v	P_v
v_1	1	0	0
v_2	1	2	v_1
update v_3	0	~~8~~ (2+6)	v_2
v_4	0	7	v_1
v_5	0	(2+2) 4	v_2

4) v_5

	known	d_v	P_v
v_1	1	0	0
v_2	1	2	v_1
v_3	0	(4+2) 6	v_5 (update)
v_4	0	(4+1) 5	v_5 (update)
v_5	1	4	v_2

5) v_4

	known	d_v	P_v
v_1	1	0	0
v_2	1	2	v_1
v_3	0	6	v_5
v_4	1	5	v_5
v_5	1	4	v_2

6) v_3

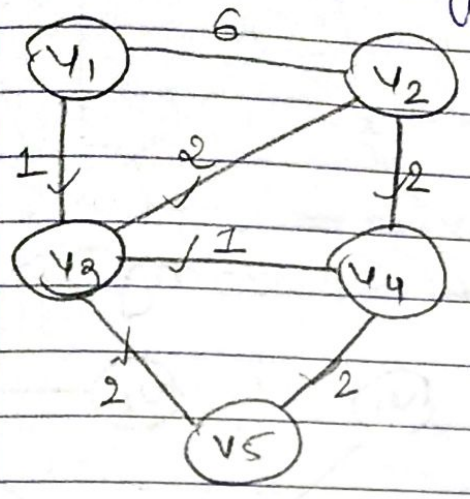
	known	d_v	P_v
v_1	1	0	0
v_2	1	2	v_1
v_3	1	6	v_5
v_4	1	5	v_5
v_5	1	4	v_2

PAGE NO.:
DATE:

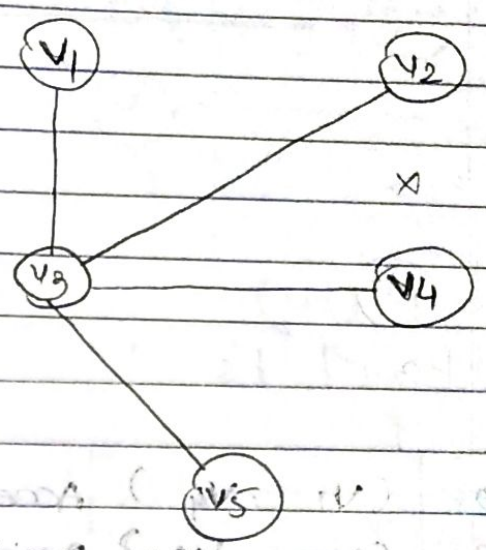
Q3 Minimum Spanning Tree :

* Kruskal's Algorithm :

(1)

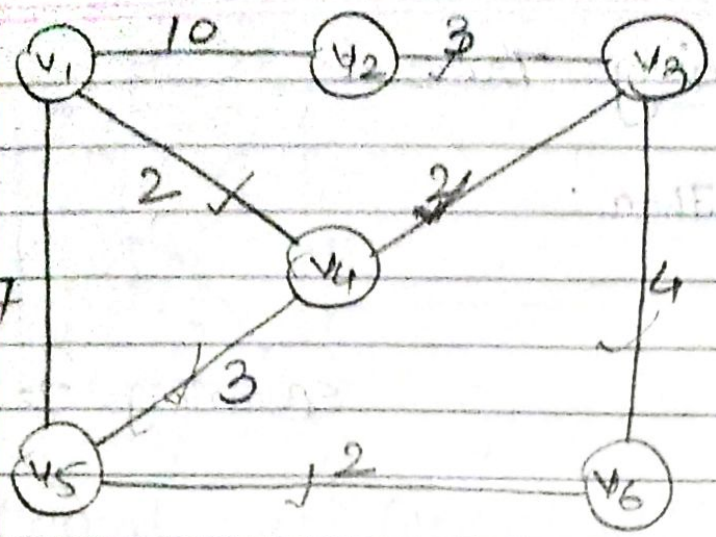


Spanning Tree :

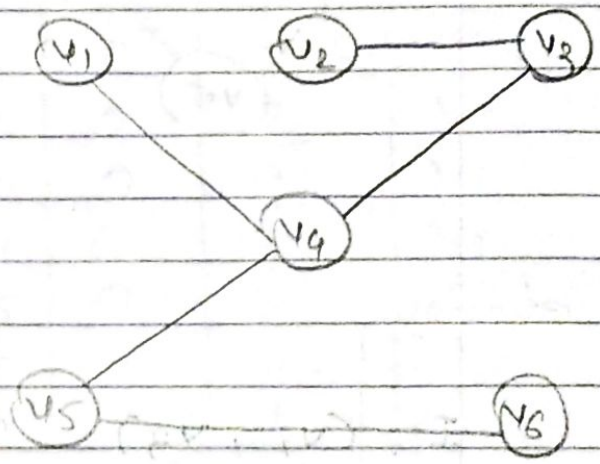


edge weight			
1	(v_1, v_3)	Accepted	(not cycled)
1	(v_3, v_4)	Accepted	
2	(v_2, v_3)	Accepted	
2	(v_2, v_4)	Rejected	(cycled)
2	(v_3, v_5)	Accepted	
2	(v_4, v_5)	Rejected	
6	(v_1, v_2)	Rejected	

2)



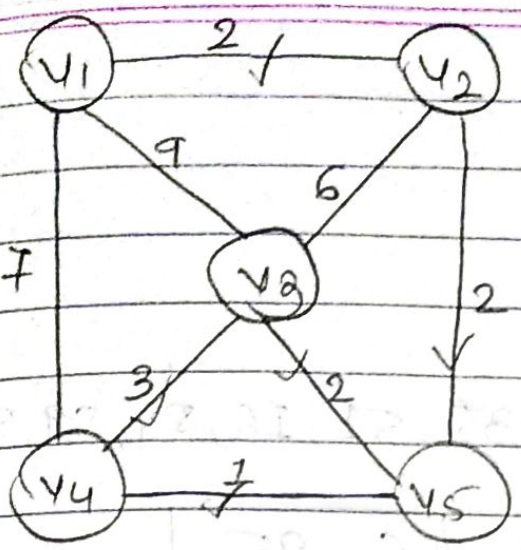
* Spanning Tree



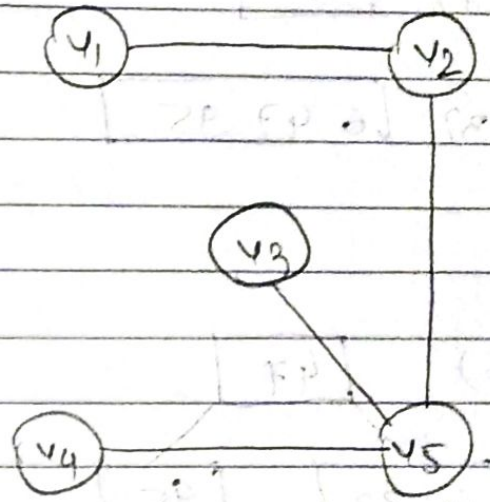
* Kruskal's :

2	(v_1 , v_4)	Accepted
2	(v_5 , v_6)	Accepted
2	(v_3 , v_4)	Accepted
3	(v_2 , v_3)	Accepted
3	(v_4 , v_5)	Accepted
4	(v_3 , v_6)	Rejected
7	(v_1 , v_5)	Rejected
10	(v_1 , v_2)	Rejected

3)



Spanning Tree:

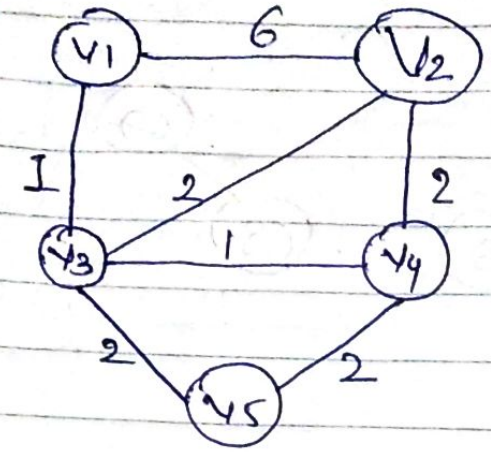


* Kruskal's

1	(v_4 , v_5)	Accepted
2	(v_1 , v_2)	Accepted
2	(v_2 , v_5)	Accepted
2	(v_3 , v_5)	Accepted
3	(v_3 , v_4)	Rejected
6	(v_2 , v_3)	Rejected
7	(v_1 , v_4)	Rejected
9	(v_1 , v_3)	Rejected

Prim's Algo :

1)



v_1 is starting node

1)

Initial state:

	known	dv	Pv
v_1	1	0	0
v_2	0	6	v_1
v_3	0	1	v_1
v_4	0	∞	0
v_5	0	∞	0

v_3

	kn	dv	Pv
v_1	1	0	0
v_2	0	~~2~~ 2	v_3
v_3	1	1	v_1
v_4	0	1	v_3
v_5	0	2	v_3

v_4

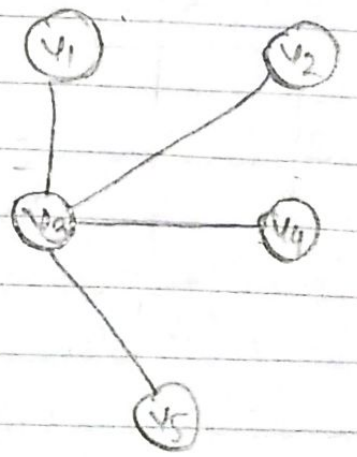
	kn	dv	
v_1	1	0	
v_2	0	2	
v_3	1	1	
v_4	1	1	
v_5	0	2	~~v_4~~

v_2

	kn	dv	Pv
v_1	1	0	0
v_2	1	2	v_3 ~~v_4~~
v_3	1	1	v_1
v_4	1	1	v_3
v_5	0	2	v_3 ~~v_4~~

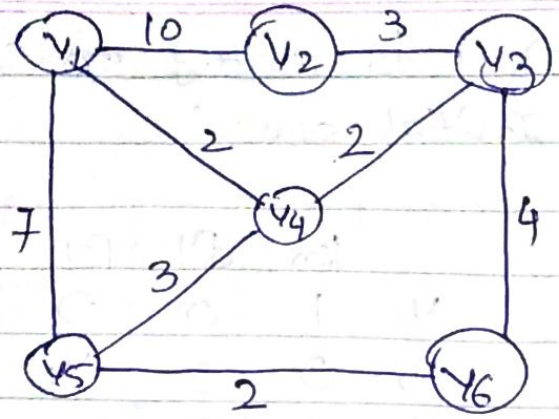
v_5

	kn	dv	Pv
v_1	1	0	0
v_2	1	2	v_3 ~~v_4~~
v_3	1	1	v_1
v_4	1	1	v_3
v_5	1	2	v_3 ~~v_4~~



minimum spanning Tree

2)



v1 is starting node.

1) Initial state

	Kn	dv	Pv
v1	1	0	0
v2	0	10	v1
v3	0	∞	0
v4	0	2	v1
v5	0	7	v1
v6	0	∞	0

v4

	Kn	dv	Pv
v1	1	0	0
v2	0	10	v1
v3	0	2	v4
v4	1	2	v1
v5	0	3	v4
v6	0	∞	0

v3

	Kn	dv	Pv
v1	1	0	0
v2	0	3	v3
v3	1	2	v4
v4	1	2	v1
v5	0	3	v4
v6	0	4	v3

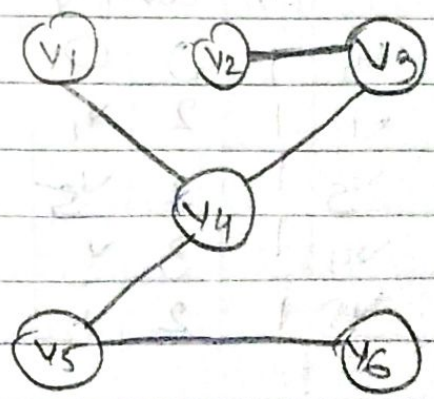
v2

	Kn	dv	Pv
v1	1	0	0
v2	1	3	v3
v3	1	2	v4
v4	1	2	v1
v5	0	3	v4
v6	0	4	v3

~~v1~~ 1

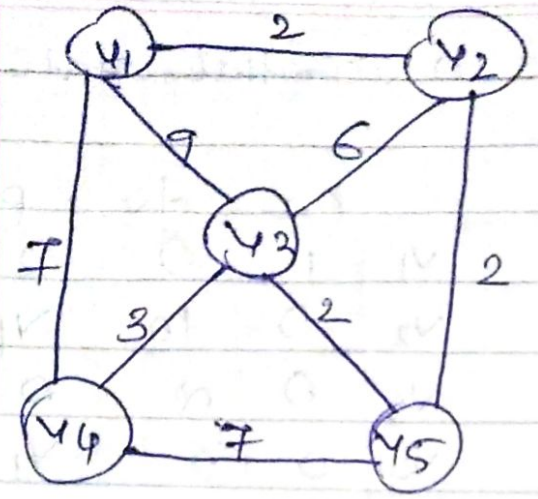
	Kn	dv	Pv
v1	1	0	0
v2	1	3	v3
v3	1	2	v4
v4	1	2	v1
v5	1	3	v4
v6	0	2	v5

	Kn	dv	Pv
v1	1	0	0
v2	1	3	v3
v3	1	2	v4
v4	1	2	v1
v5	1	3	v4
v6	1	2	v5



minimum spanning Tree

(3)



→ v_1 is starting node

Initial state:

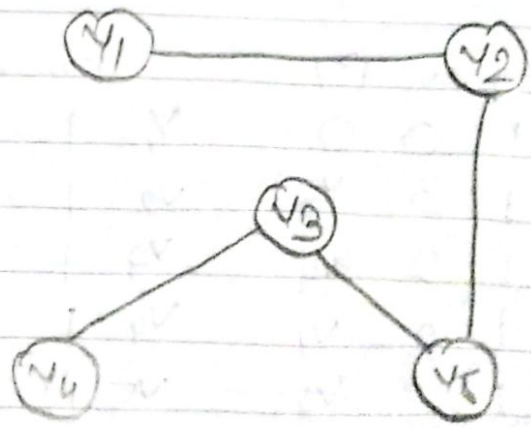
	kv	dv	Pv
v_1	1	0	0
v_2	0	2	v_1
v_3	0	9	v_1
v_4	0	7	v_1
v_5	0	∞	0

	kv	dv	Pv
v_1	1	0	0
v_2	1	2	v_1
v_3	0	6	v_2
v_4	0	7	v_1
v_5	0	2	v_2

	kv	dv	Pv
v_1	1	0	0
v_2	1	2	v_1
v_3	0	2	v_5
v_4	0	7	v_1
v_5	1	2	v_2

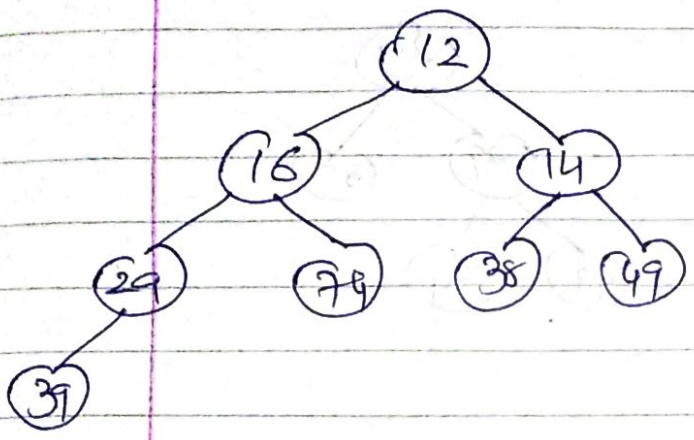
	kv	dv	Pv
v_1	1	0	0
v_2	1	2	v_1
v_3	1	2	v_5
v_4	0	3	v_3
v_5	1	2	v_2

	kv	dv	Pv
v_1	1	0	0
v_2	1	2	v_1
v_3	1	2	v_5
v_4	1	3	v_3
v_5	1	2	v_2

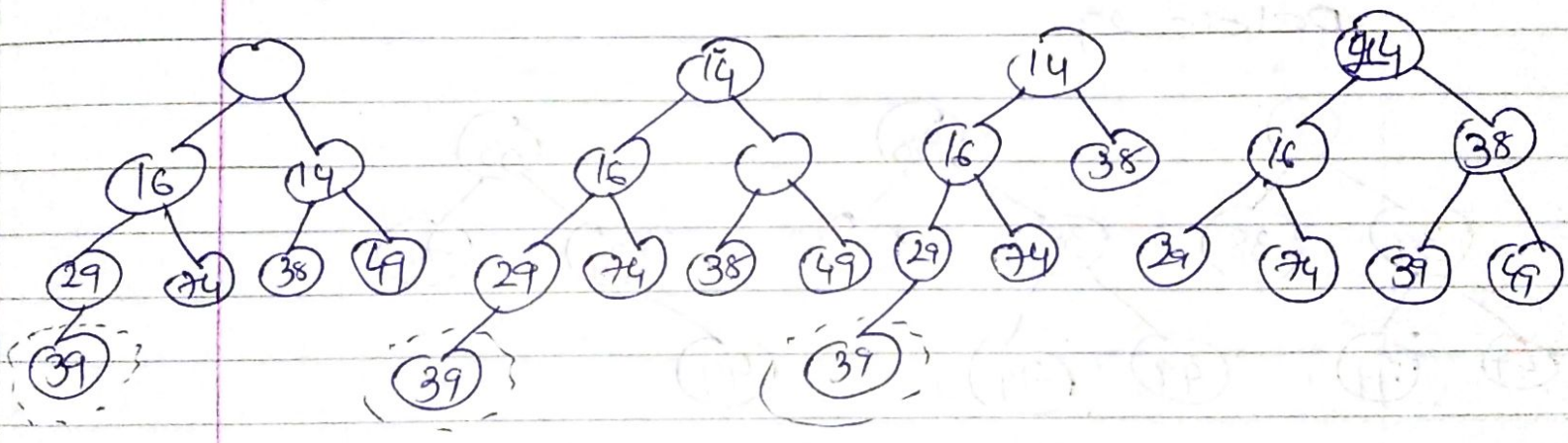


Heap sort.

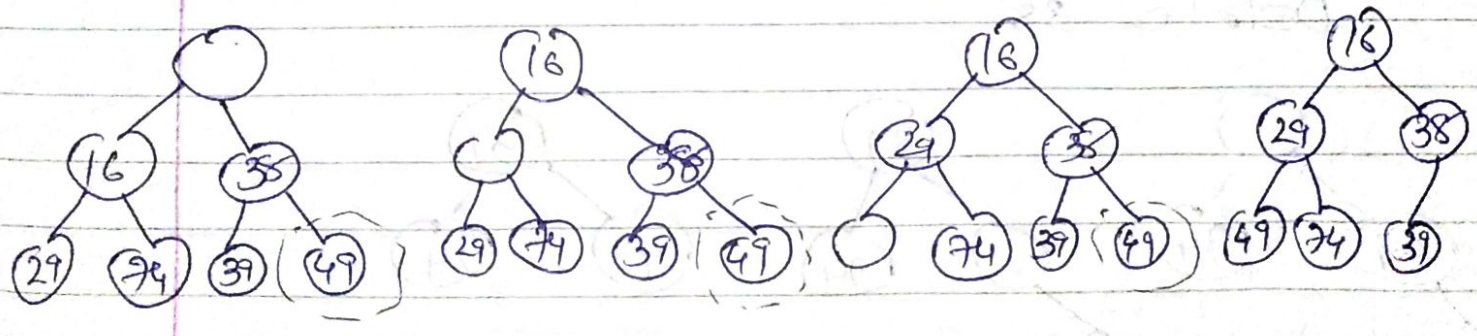
Delete min.



Step I: Delete 12.

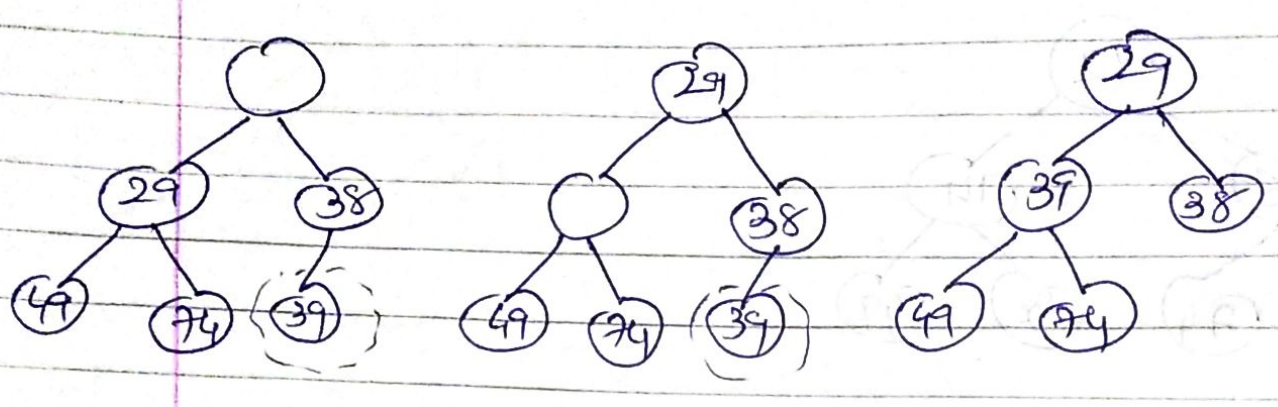


Step 2: Delete 14.



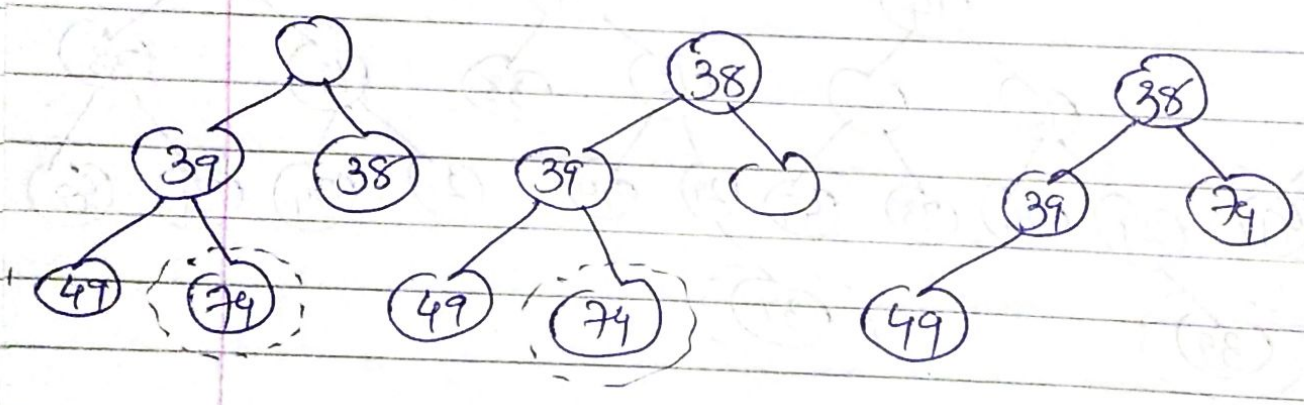
Step 3 :

Delete 16



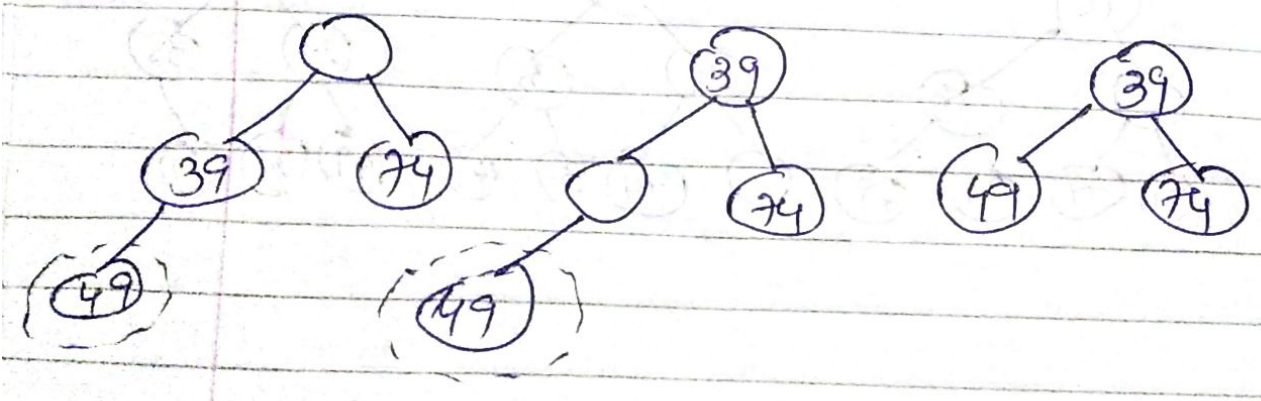
Step. 4

Delete 29

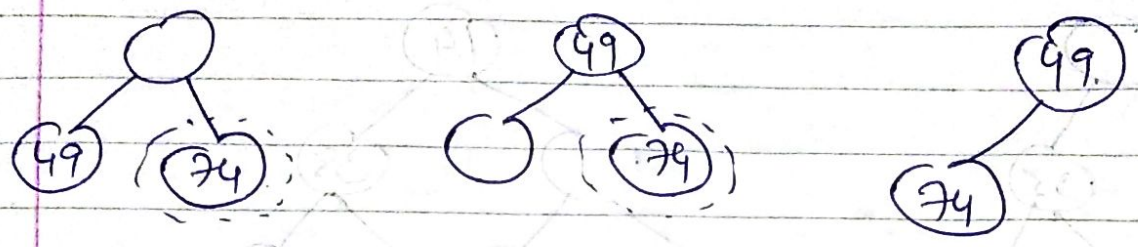


Step 5 :

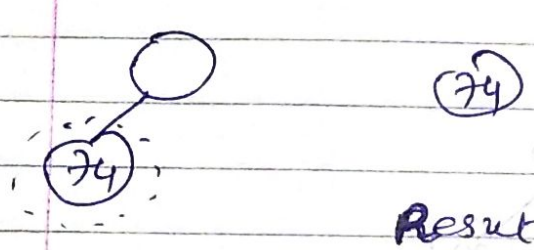
Delete 38.



Step 6 :
Delete 39.

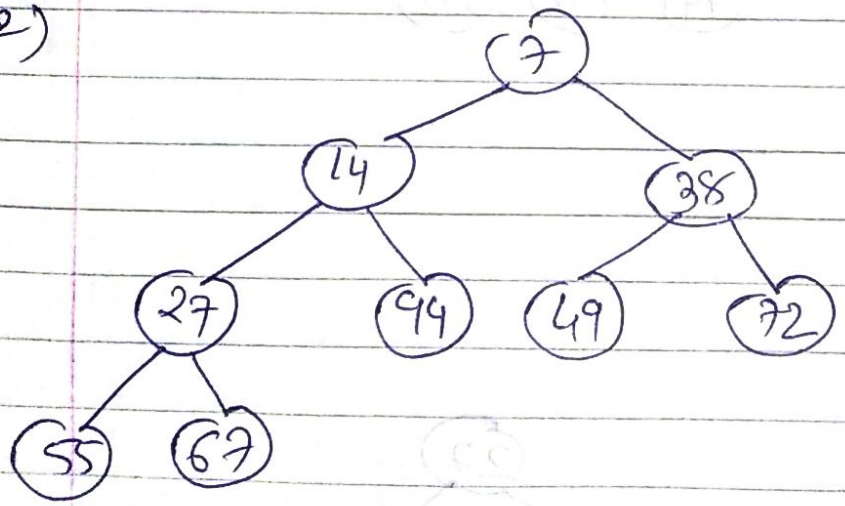


Step 7 :
Delete 49.



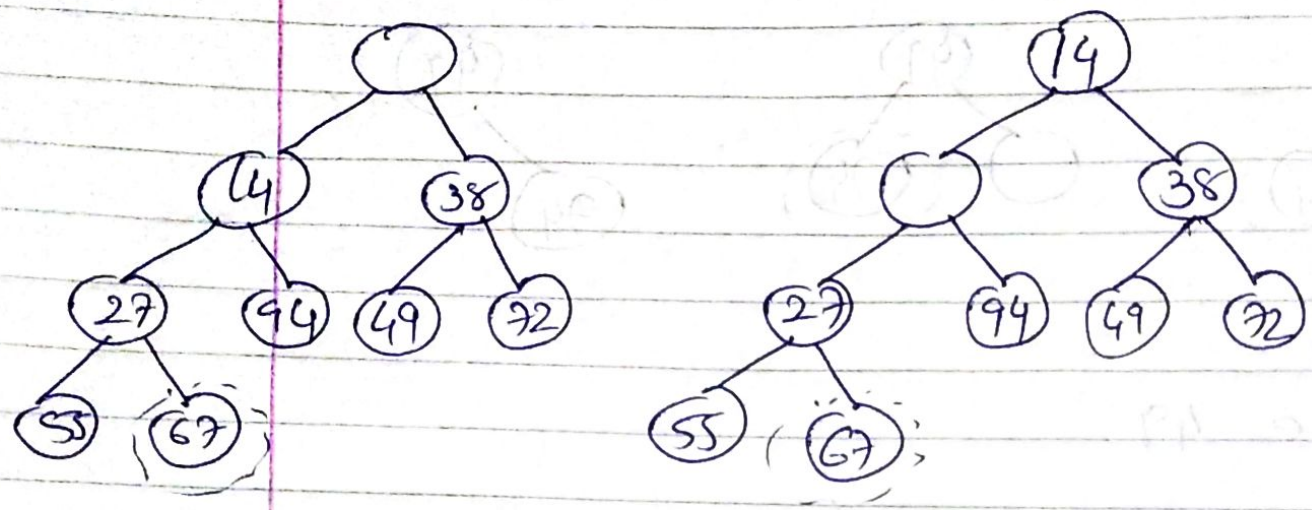
Result : 12, 14, 16, 29, 38, 39, 49, 74

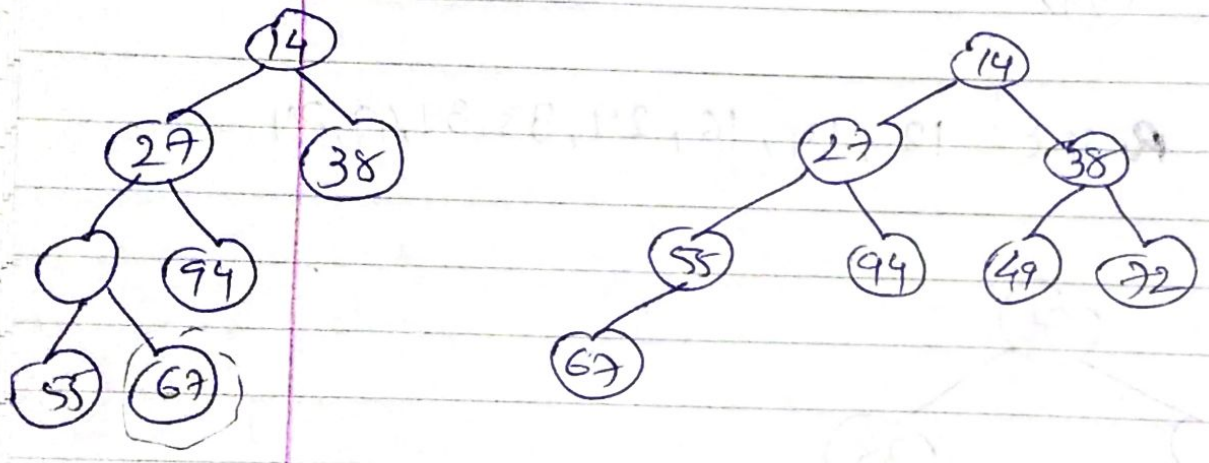
2)



Step. 1 :

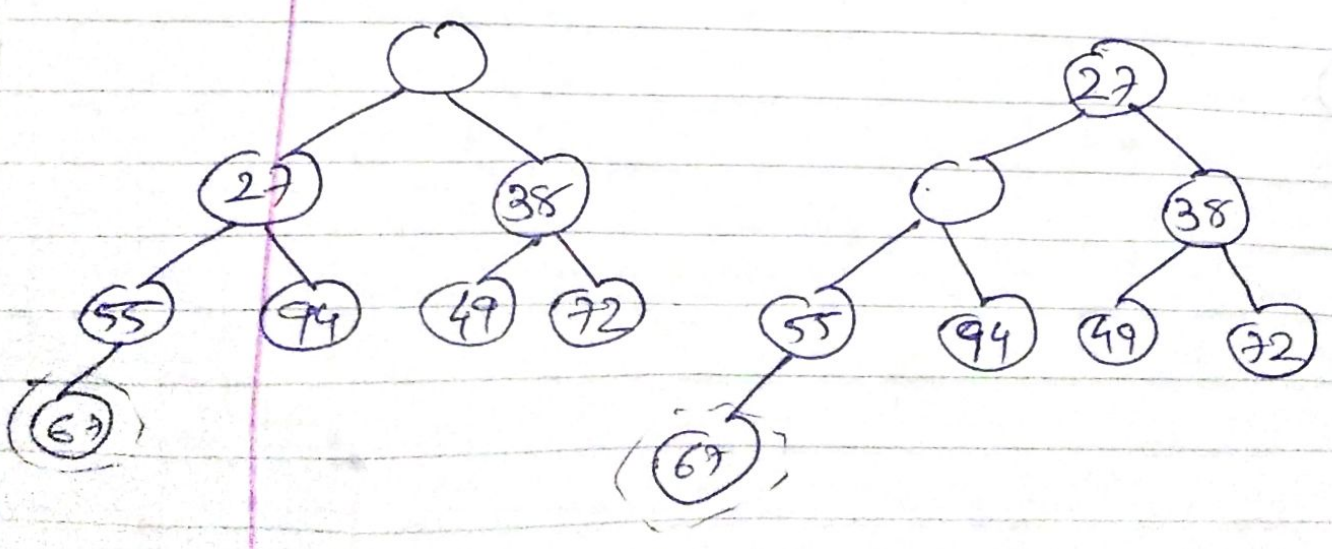
Delete 7 :

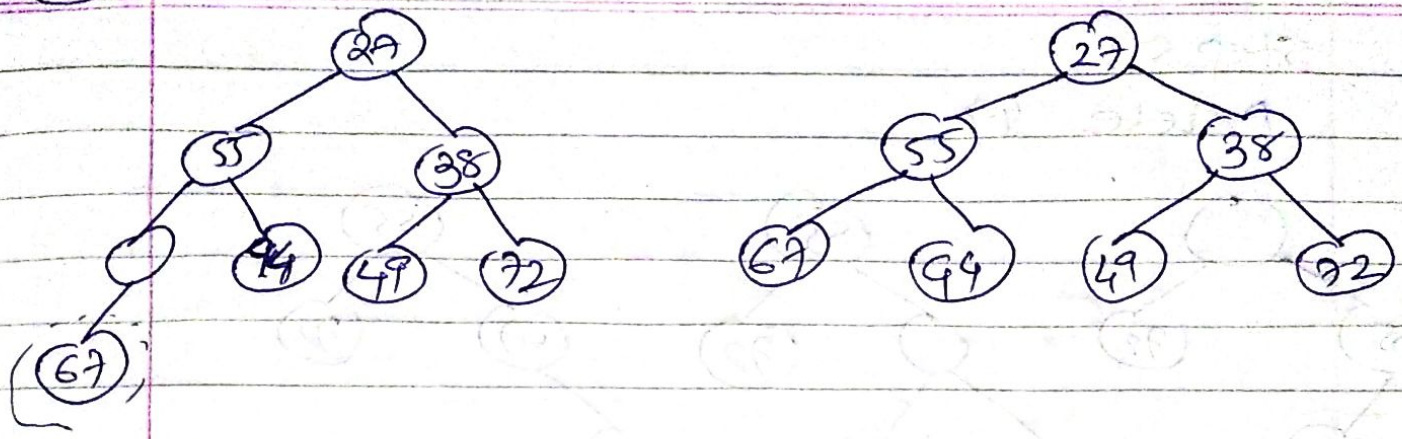




Step 2 :

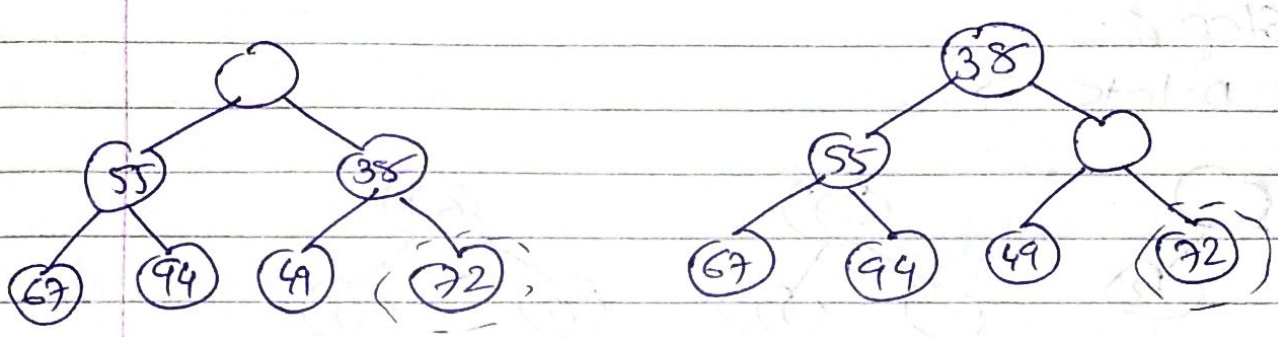
Delete 14

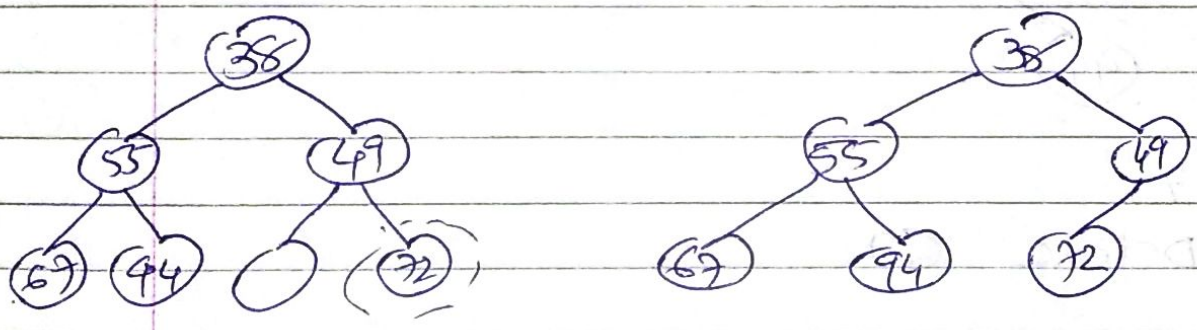




Step 3 :

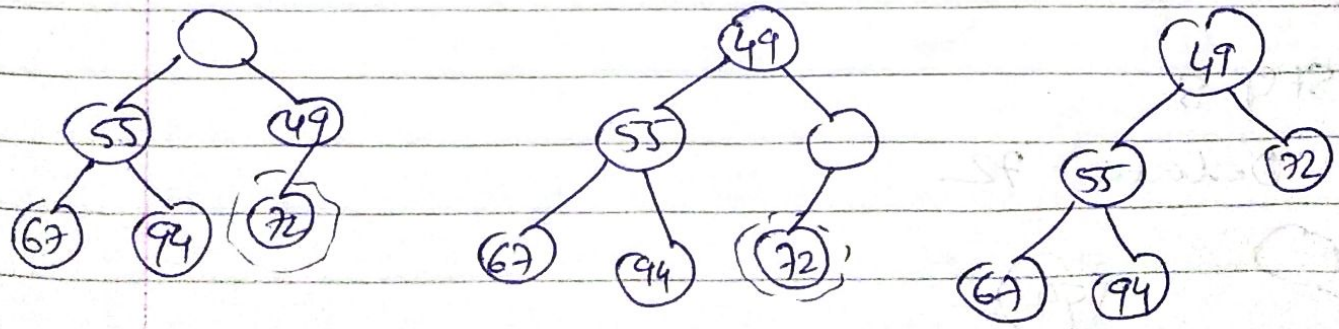
Delete 27





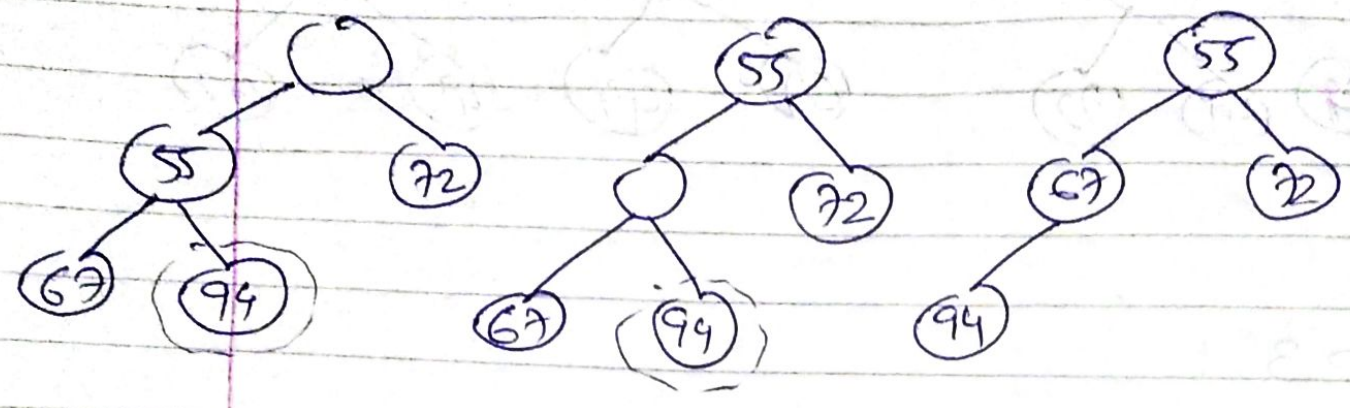
Step 4:

Delete 38



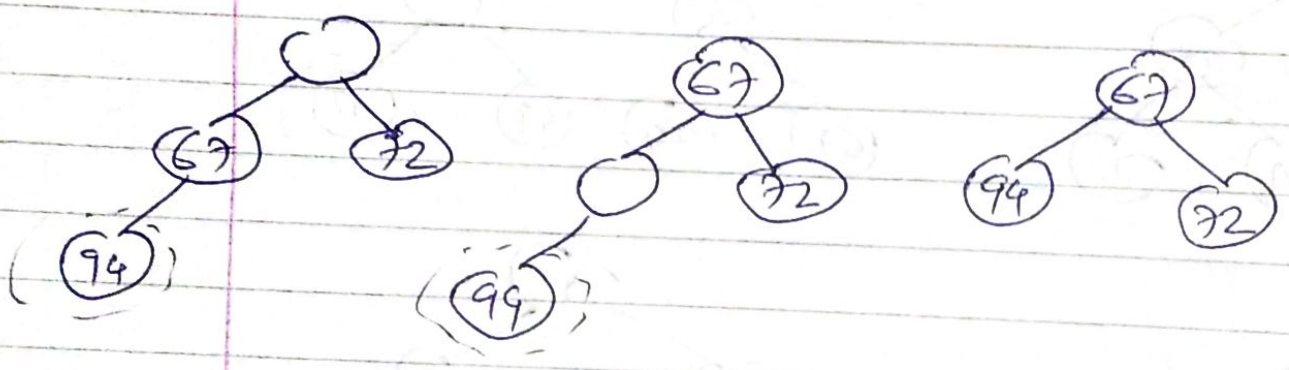
Step 5:

Delete 49



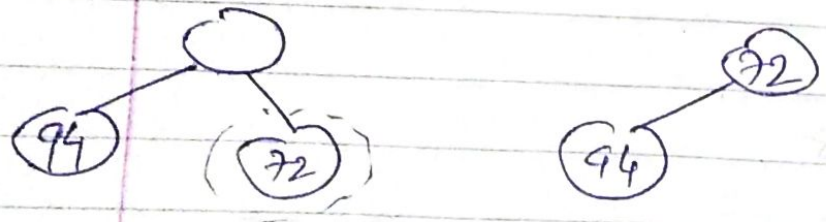
Step 6:

Delete 55.



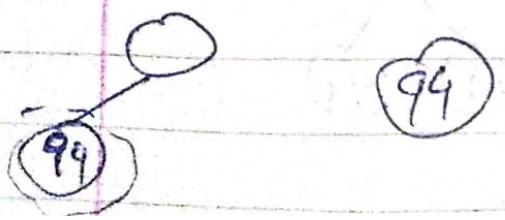
Step 7:

Delete 67



Step 8

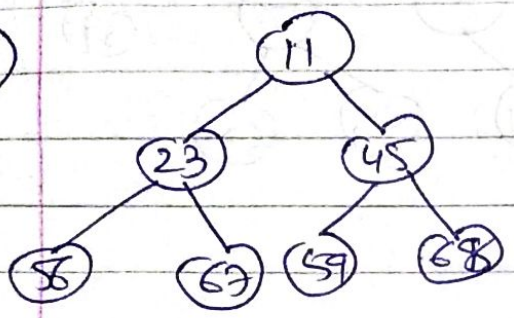
Delete 72



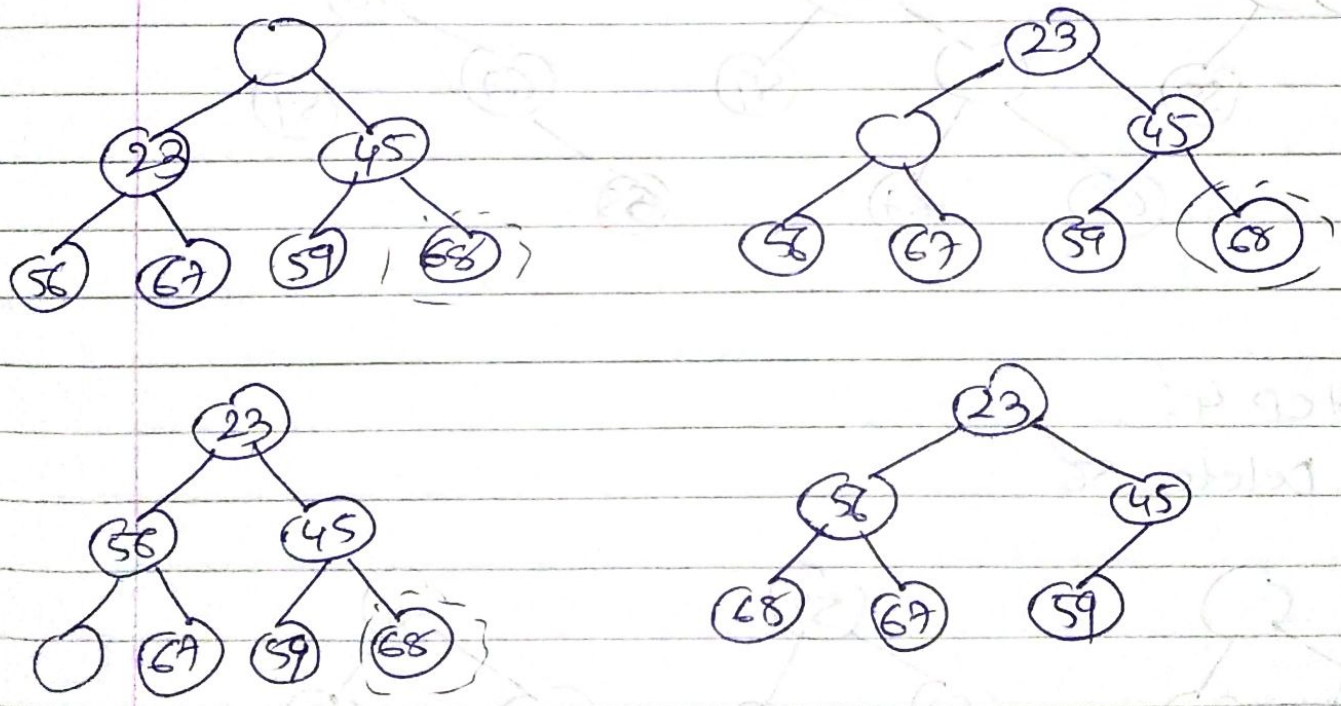
Step 9:
Delete 94

result: 7 14 21 38 49 55 67 72 94.

3)

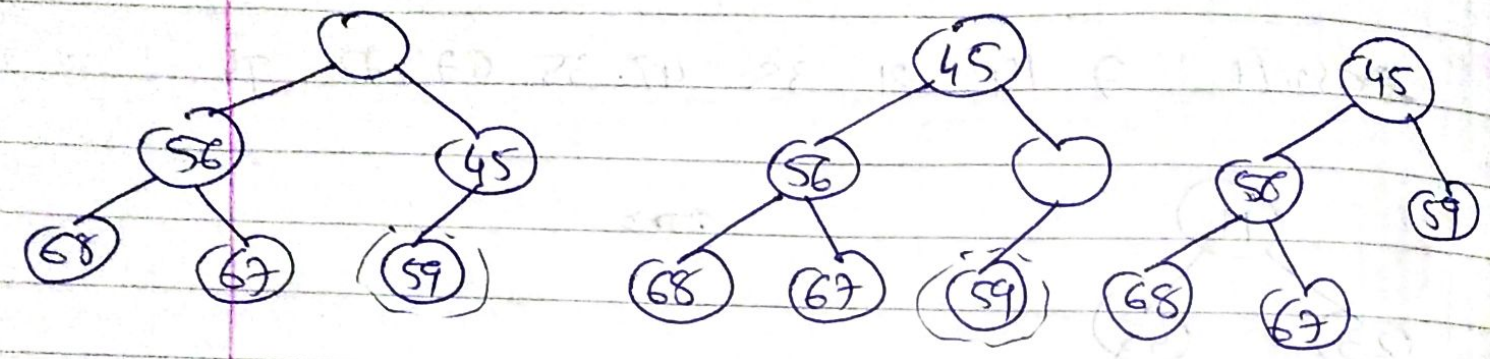


Step 1:



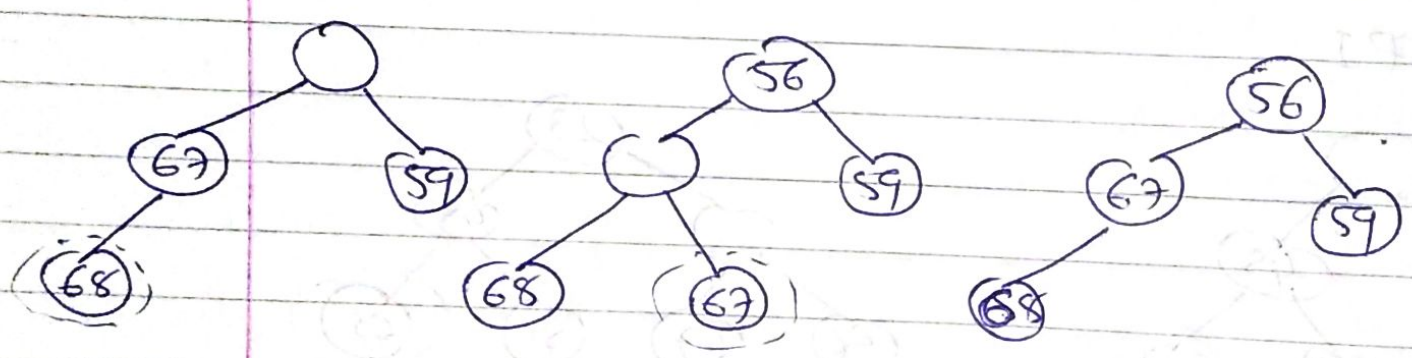
Step 2:

Delete 23



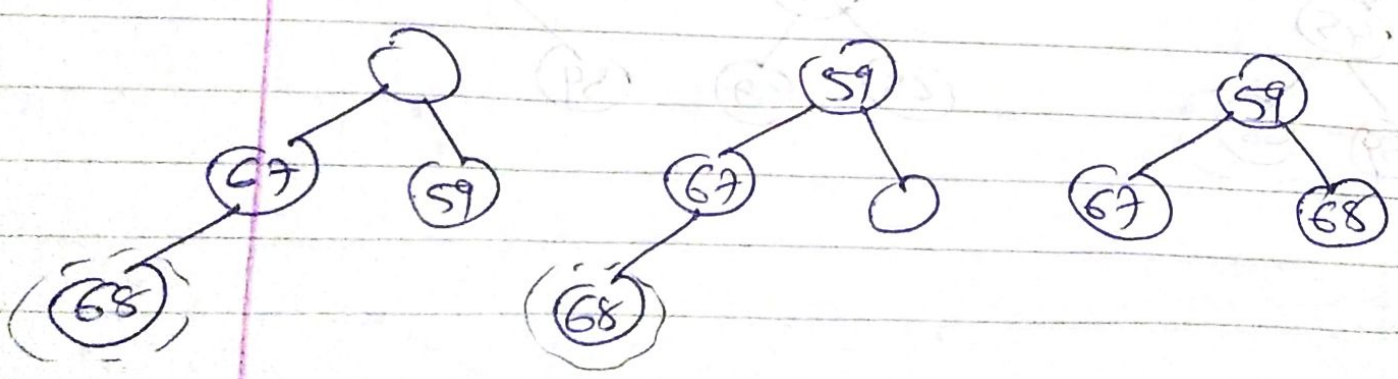
Step 3:

Delete 45



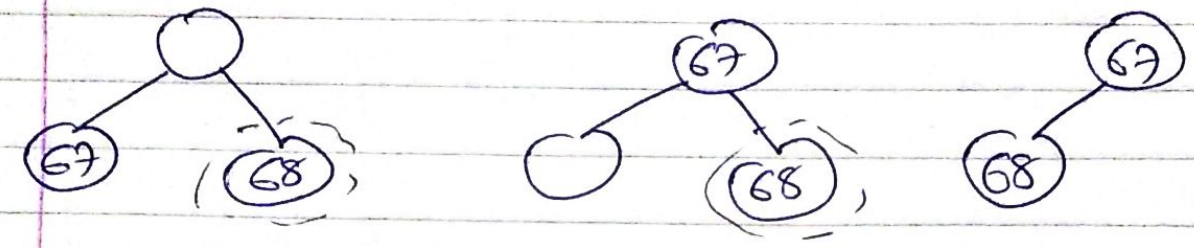
Step 4:

Delete 56



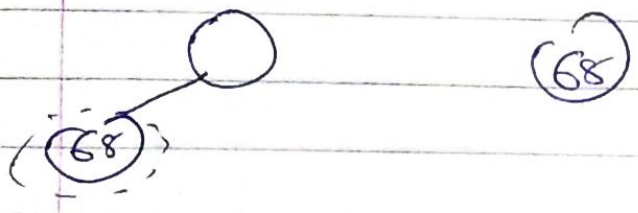
Step 5:

Delete 59



Step 6:

Delete 67



Step 7:

Delete 68

Result: 11 23 45 56 59 67 68

Assignment - 3

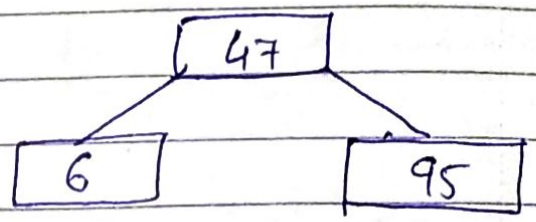
* B-Tree : order 3

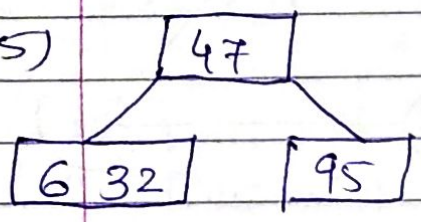
6, 95, 47, 32, 28, 75, 87, 35, 51, 16, 89, 89, 97, 99

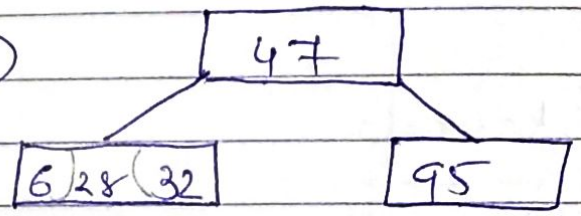
1) [6]

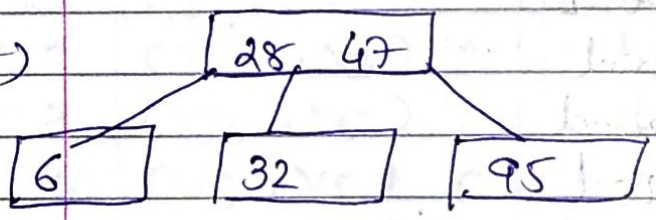
2) [6 95]

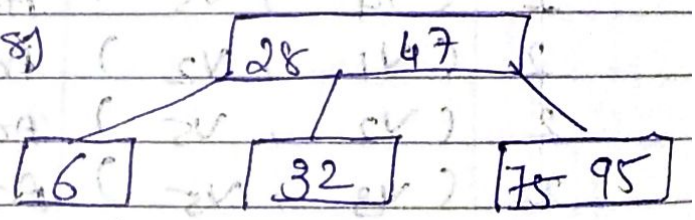
3) [6 47 95]

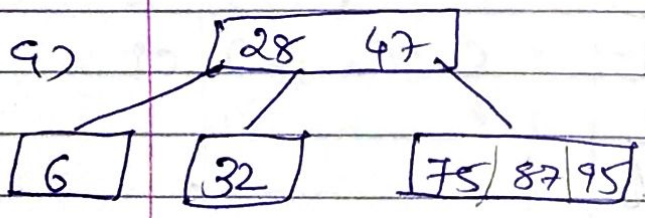
4)


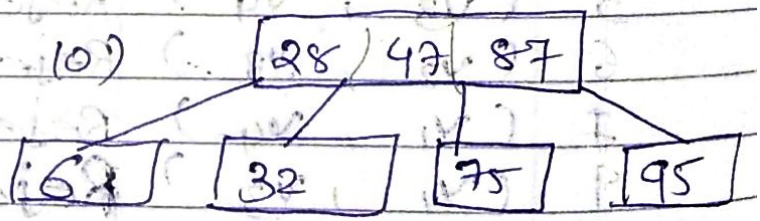
5)


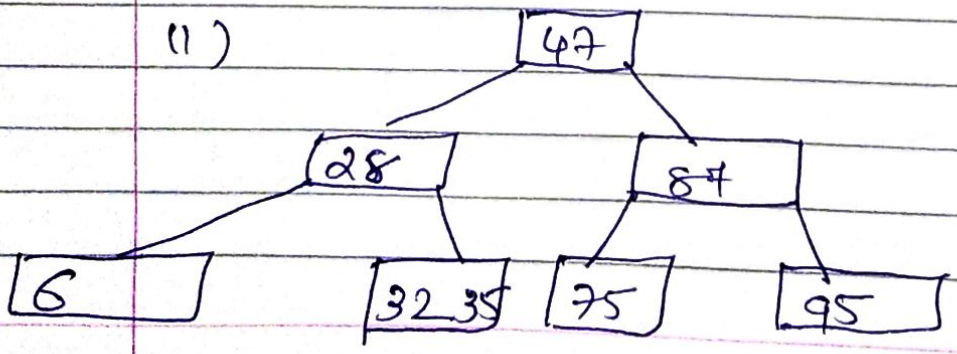
6)


7)


8)


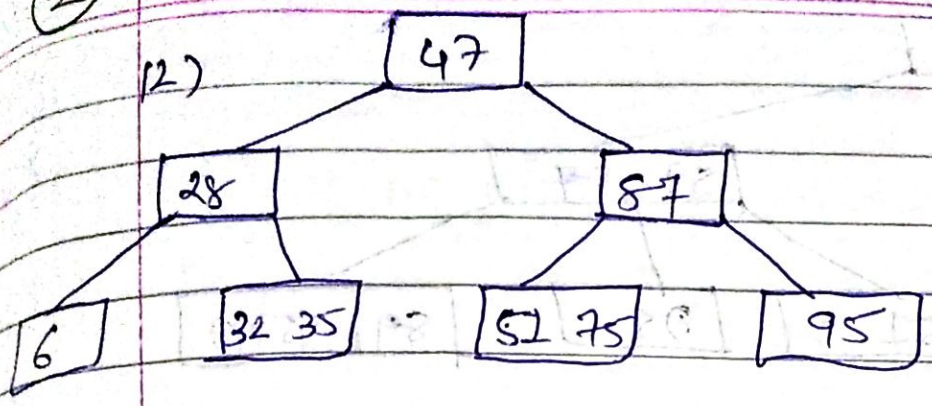
9)


10)


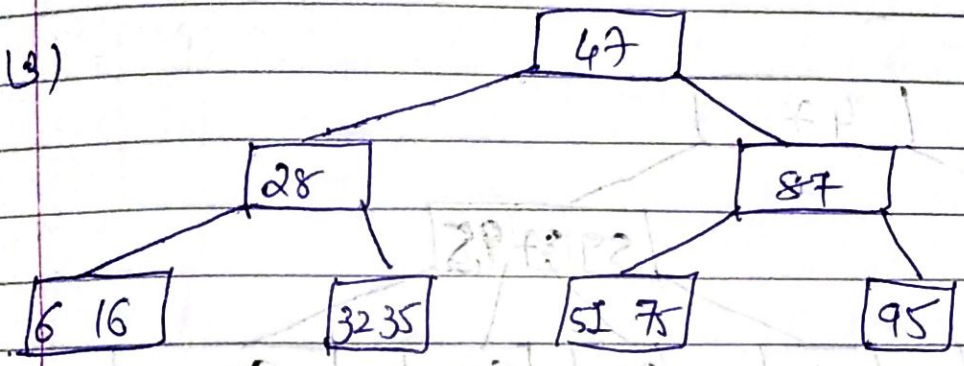
11)


②

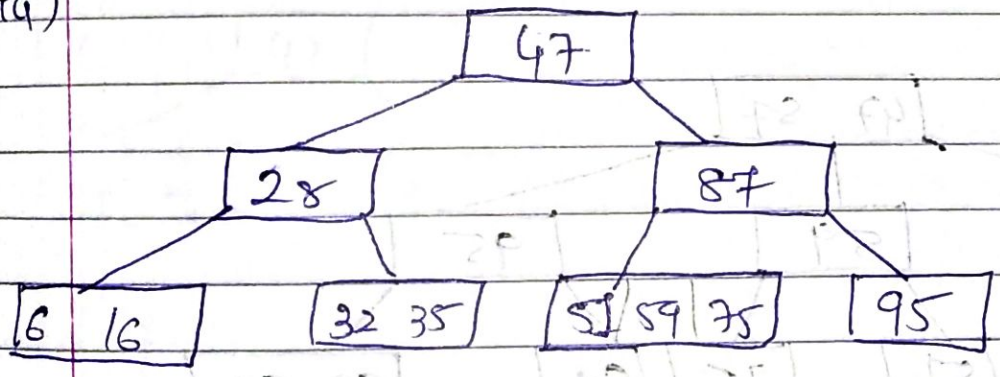
(2)


47
28
87
6
32 35
51 75
95

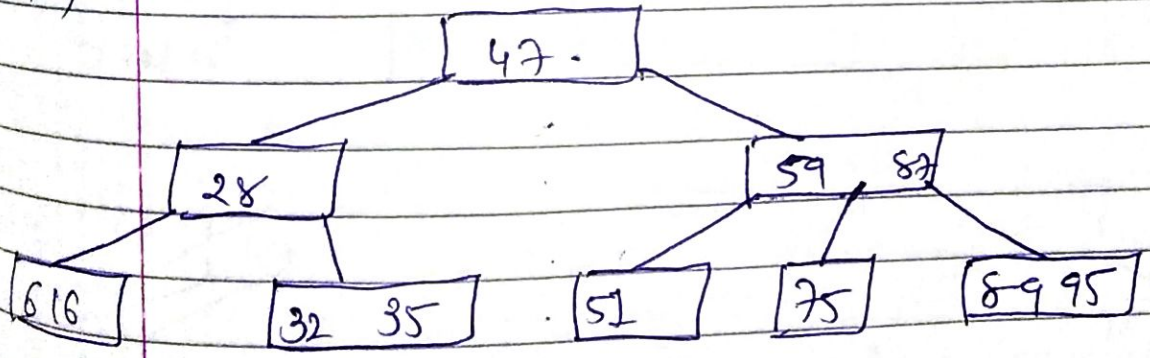
(3)


47
28
87
6 16
32 35
51 75
95

(4)


47
28
87
6 16
32 35
51 59 75
95

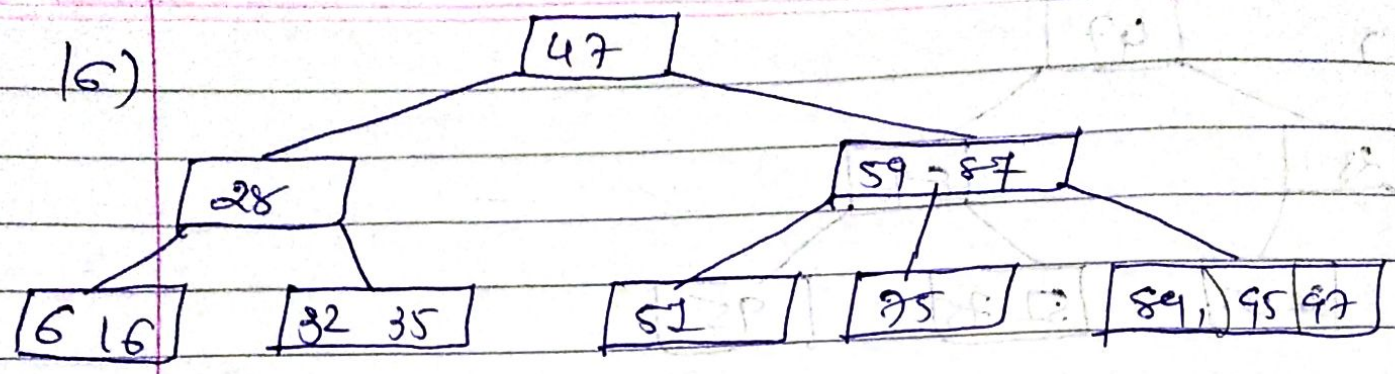
(5)


47
28
59 87
6 16
32 35
51
75
89 95

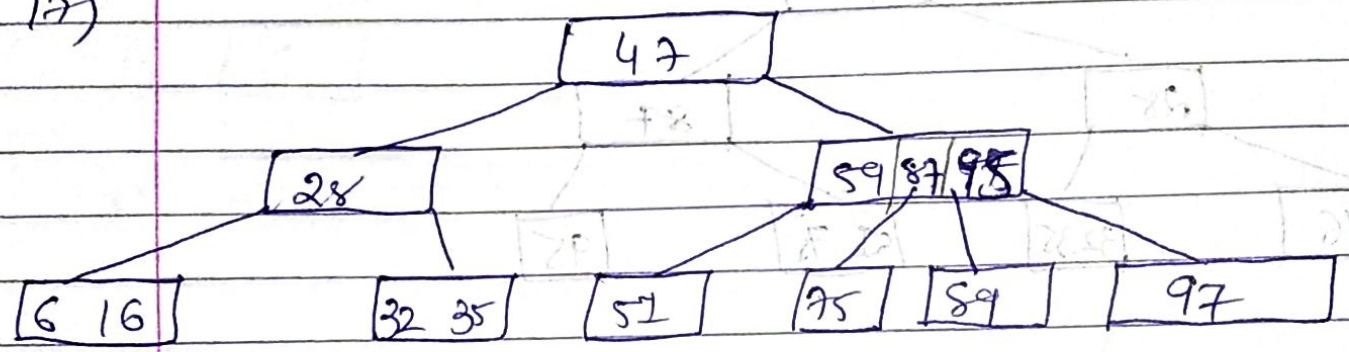
(6)

③

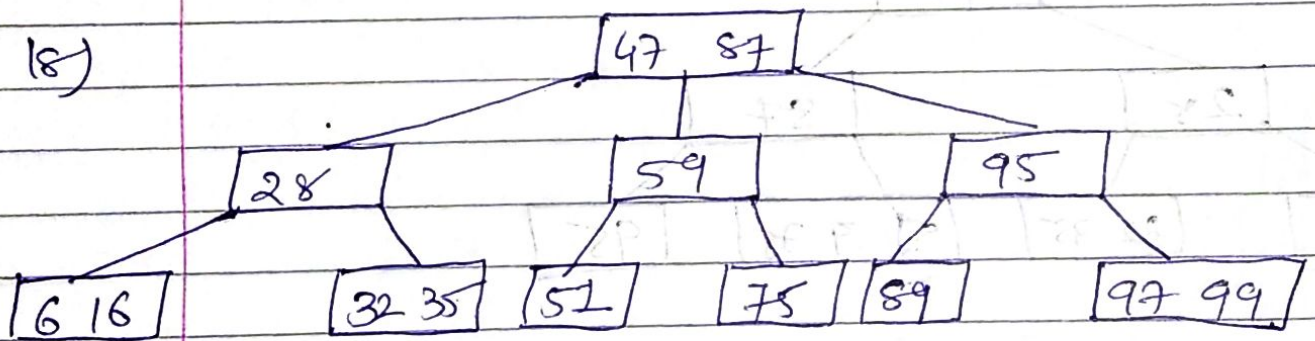
16)


47
28
59 87
6 16
32 35
51
75
89
95
97

17)


47
28
59 87 95
6 16
32 35
51
75
89
97

18)


47 87
28
59
95
6 16
32 35
51
75
89
97 99

4

order 4:

6, 95, 47, 32, 28, 75, 87, 35, 51, 16, 59, 89, 97, 99

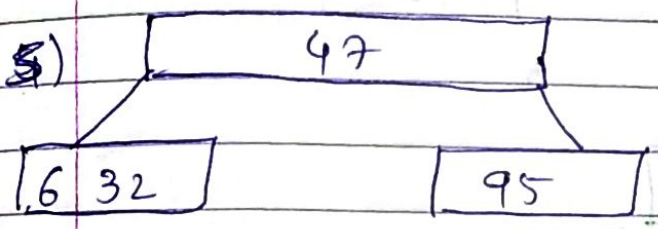
1) [6]

2) [6 95]

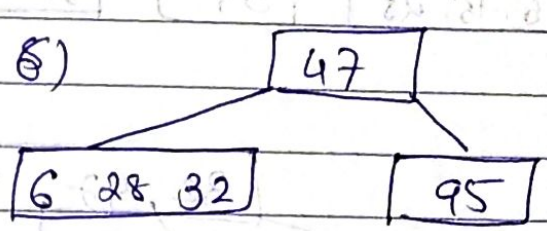
3) [6 47 95]

4) [6 32 | 47 | 95]

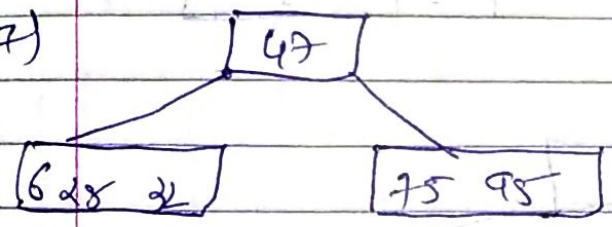
5)



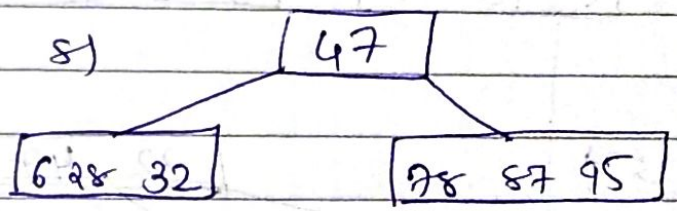
6)



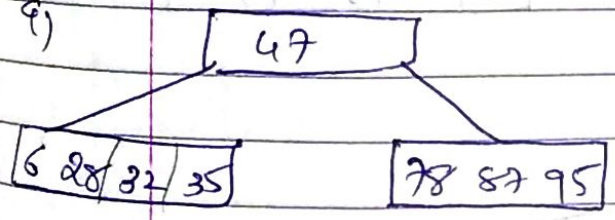
7)



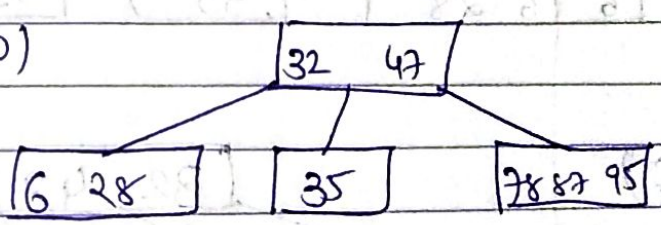
8)



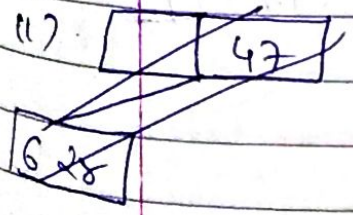
9)



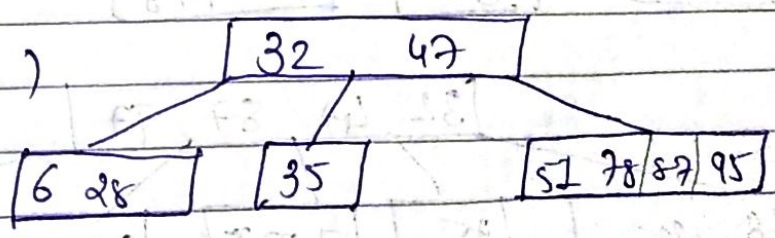
10)



11)

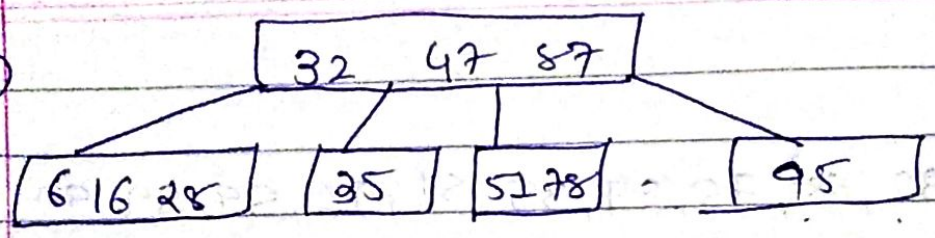


11)

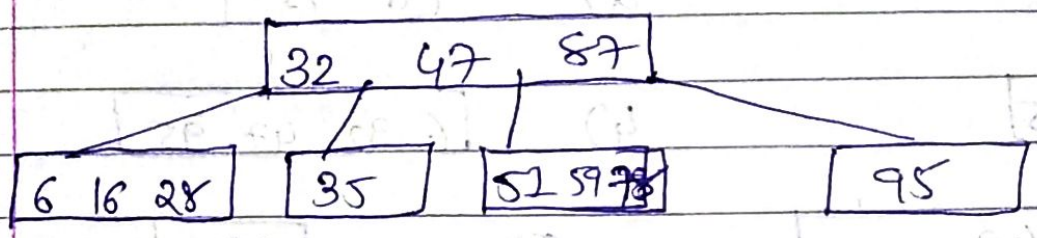


DELUXE
PAGE NO.:
DATE:

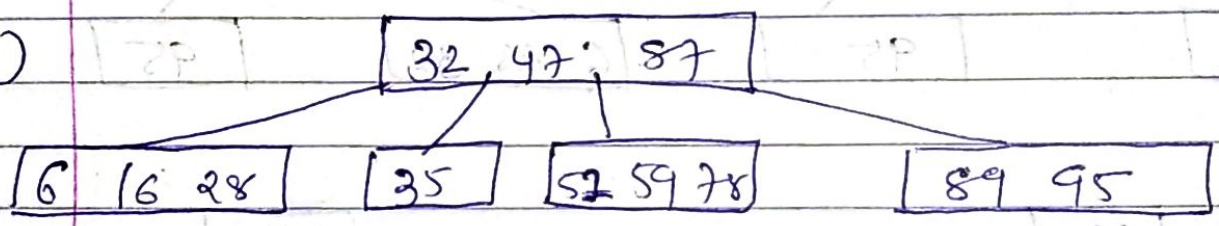
(2)



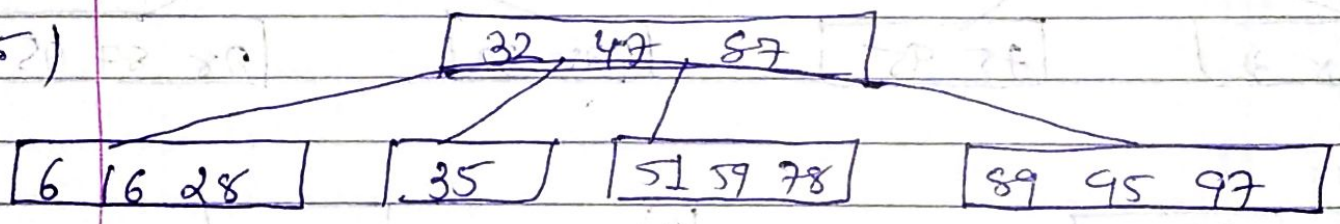
(3)



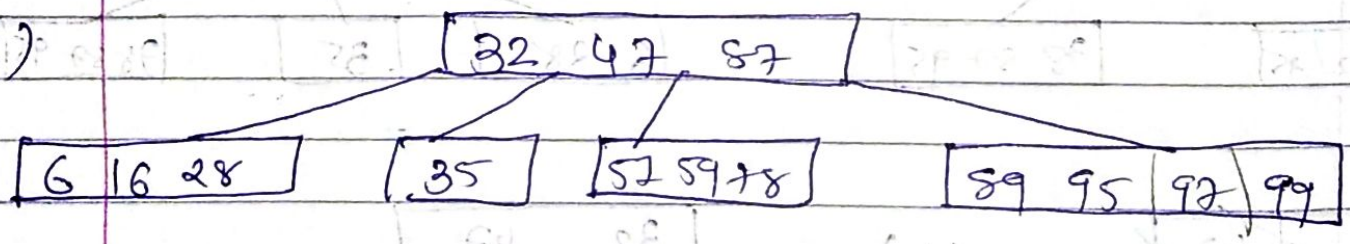
(4)



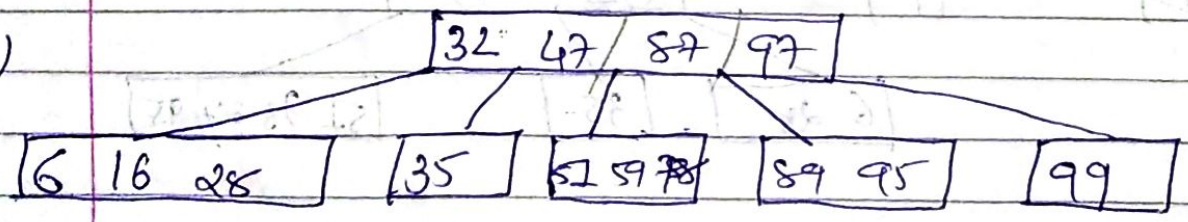
(5)



(6)

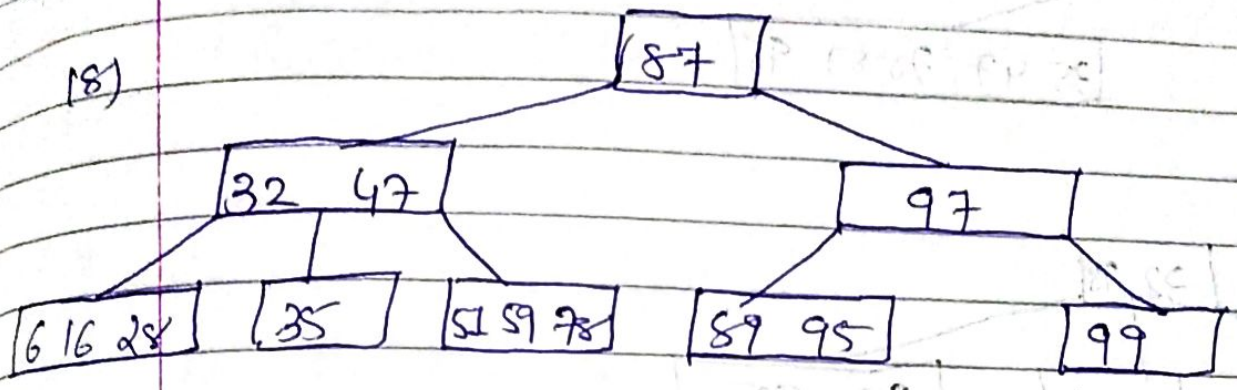


(7)



(6)

(8)



Order-5

6, 95, 47, 32, 28, 75, 87, 35, 51, 16, 59, 89, 97, 99

1) [6]

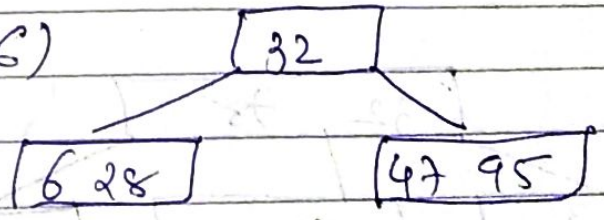
2) [6 95]

3) [6 47 95]

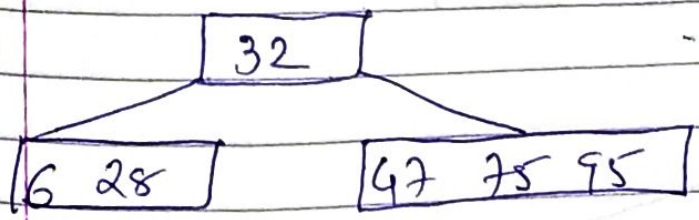
4) [6 32 47 95]

5) [6 28 32 47 95]

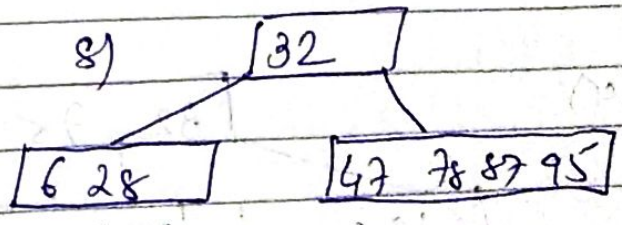
6)



7)

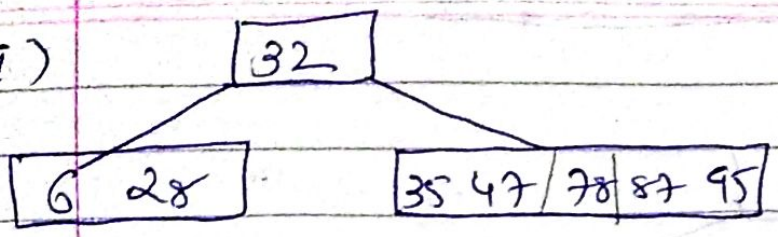


8)

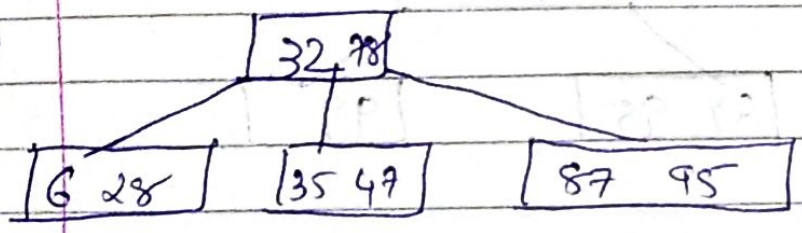


7

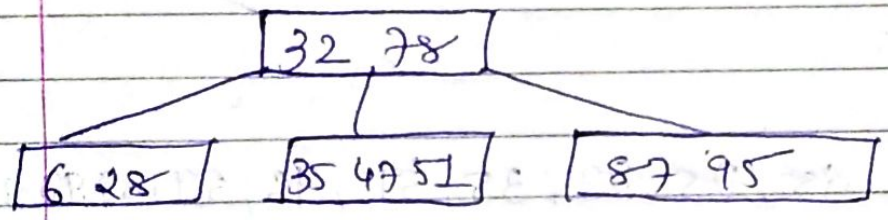
9)



10)

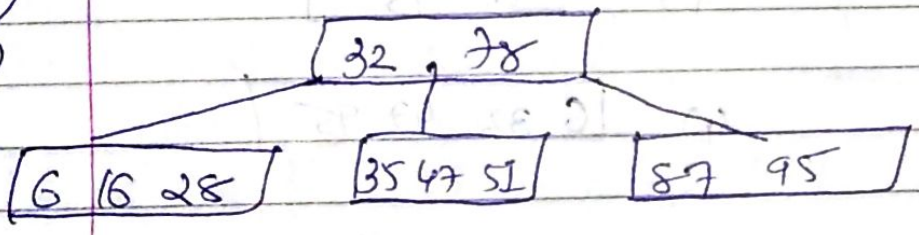


11)

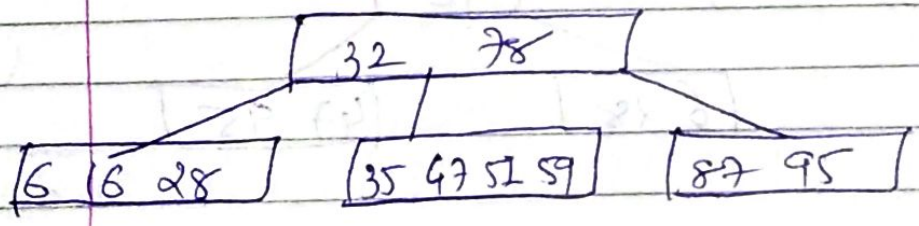


~~12)~~

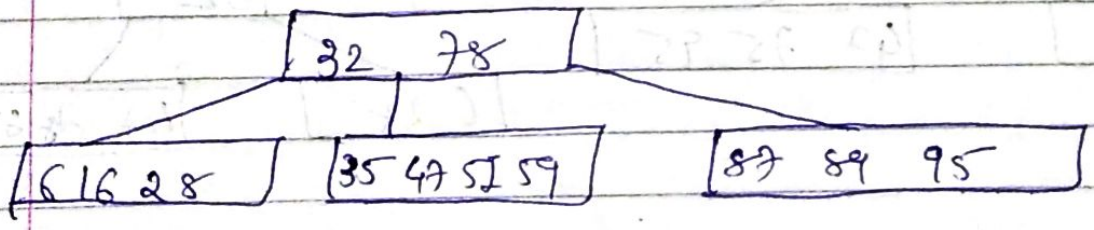
12)



13)

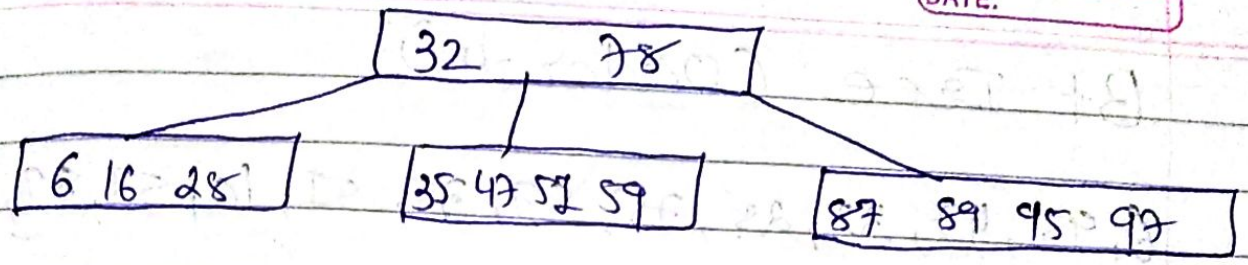


14)

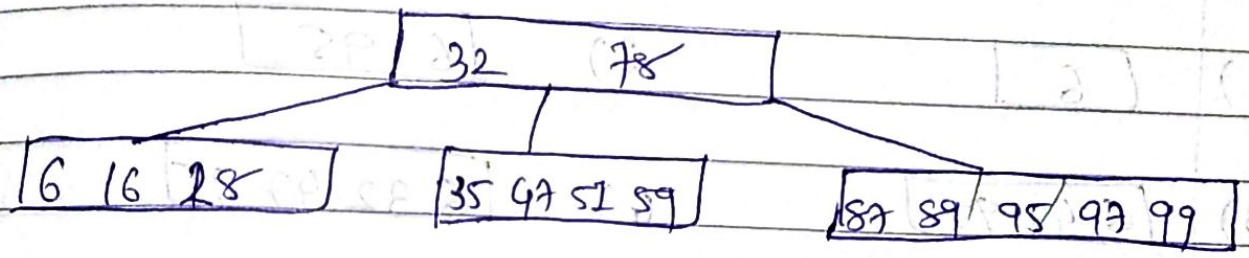


(4)

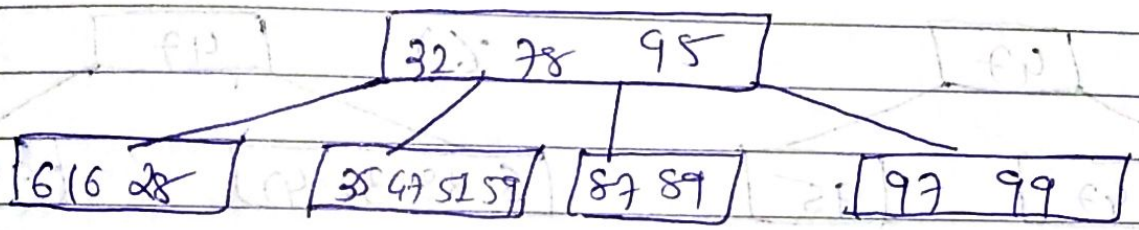
15)


32 78
6 16 28
35 47 51 59
87 89 95 97

(6)


32 78
6 16 28
35 47 51 59
87 89 95 97 99

(7)


32 78 95
6 16 28
35 47 51 59
87 89
97 99

B+ Tree (Order-4)

~~6, 95, 47, 32, 28, 75, 87, 35, 51, 16, 54, 89, 97, 99.~~

1) [6]

2) [6 | 95]

3) [6 47 95]

4) [6 32 / 47 / 95]

5)
- [47]
  - [6 32 47]
  - [95]

6)
- [47]
  - [6 28 / 32 / 47]
  - [95]

7)
- [32 47]
  - [6 28 32]
  - [47]
  - [75 95]

8)
- [32 47]
  - [6 28 32]
  - [47]
  - [75 87 95]

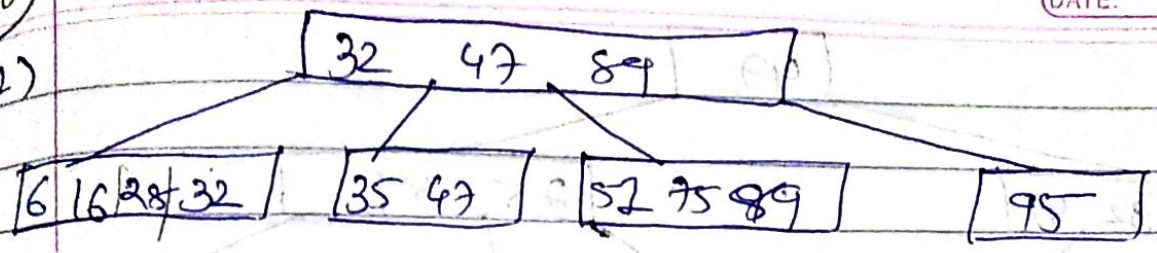
9)
- [32 47]
  - [6 28 32]
  - [35, 47]
  - [75 87 95]

10)
- [32 47]
  - [6 28 32]
  - [35 47]
  - [51 75 87 95]

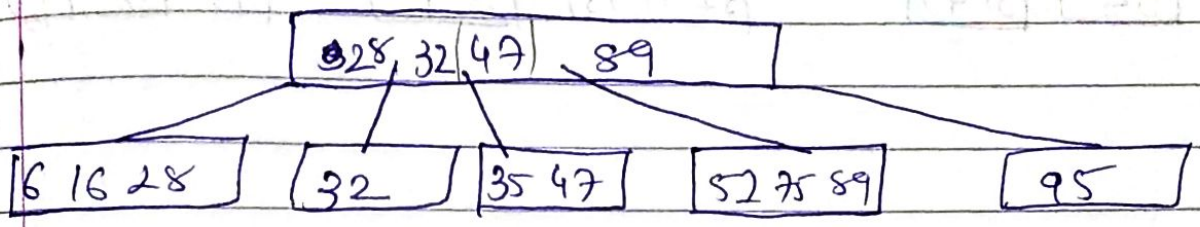
11)
- [32 47 87]
  - [6 28 32]
  - [35 47]
  - [51 75 87]
  - [95]

(10)

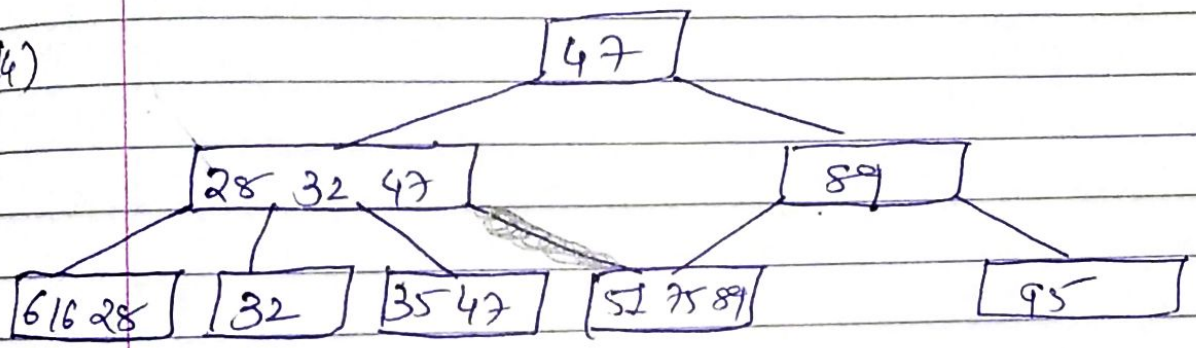
(2)


32 47 89
6 16 28 32
35 47
51 75 89
95

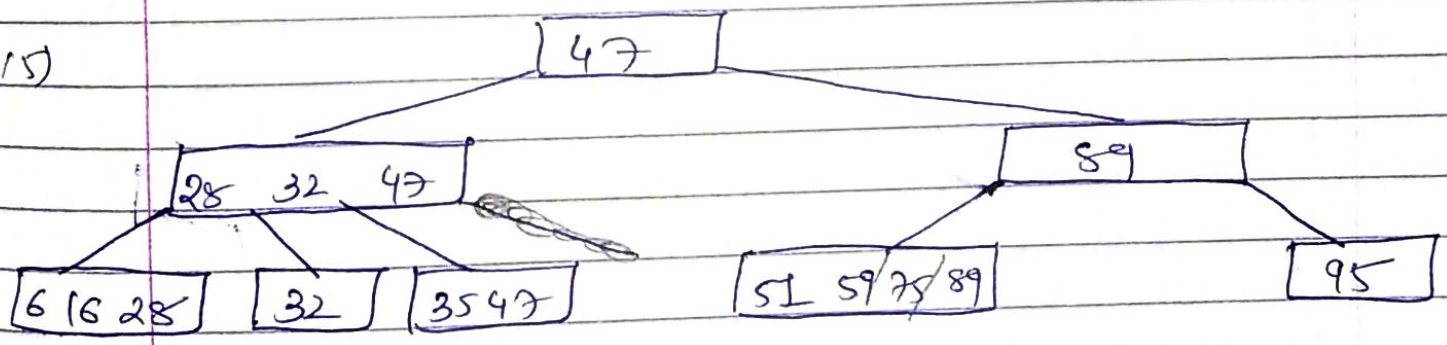
(3)


28 32 47 89
6 16 28
32
35 47
51 75 89
95

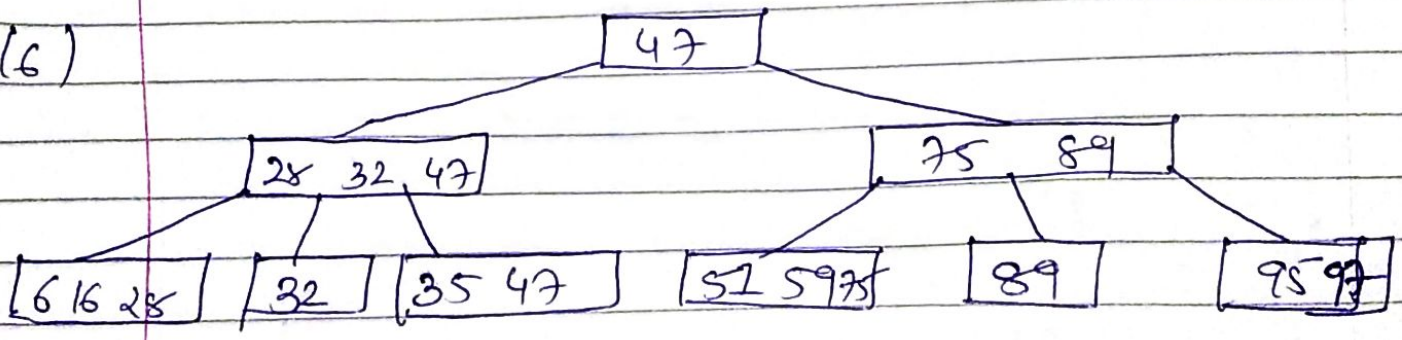
(4)


47
28 32 47
89
6 16 28
32
35 47
51 75 89
95

(5)

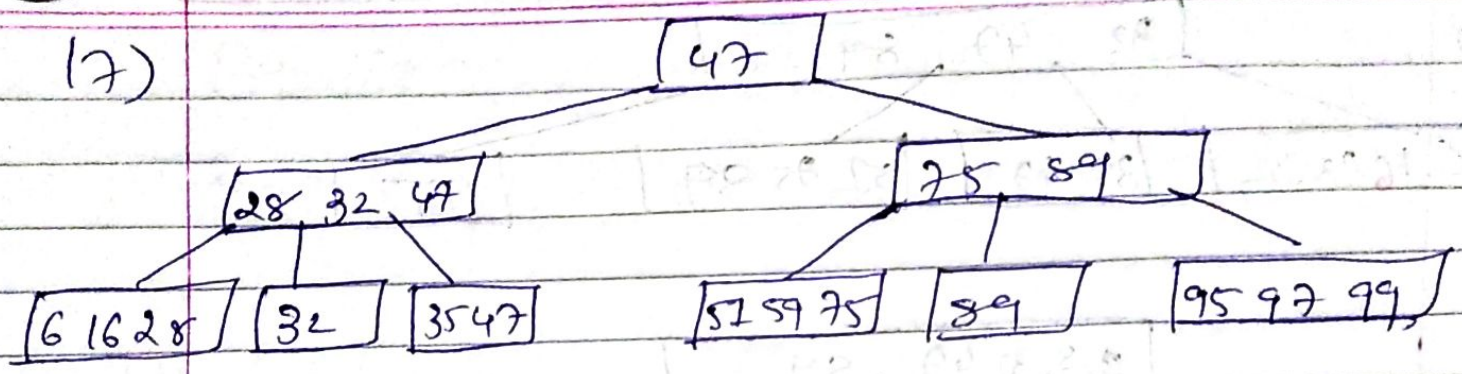

47
28 32 47
89
6 16 28
32
35 47
51 59 75 89
95

(6)


47
28 32 47
75 89
6 16 28
32
35 47
51 59 75
89
95 97

(11)

(7)



Page
Date

B* Tree (order 4)

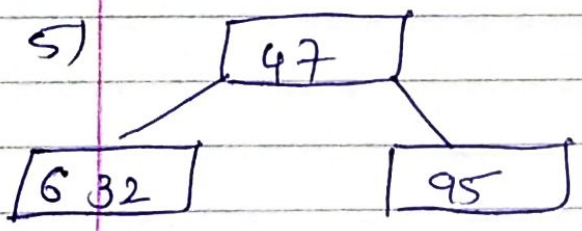
6, 95, 47, 32, 28, 75, 87, 35, 52, 16, 59, 89, 77, 99.

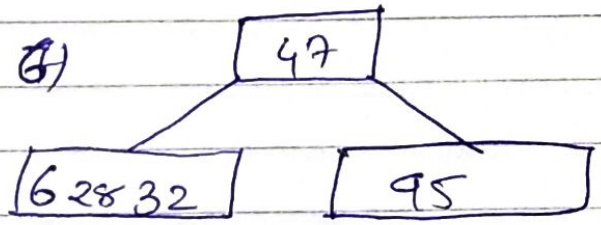
1) [6]

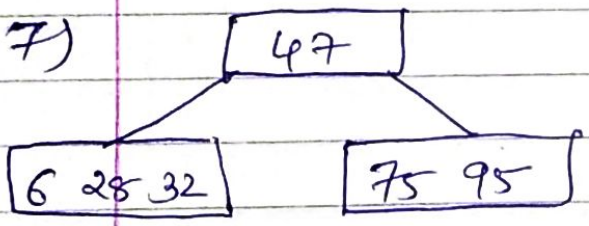
2) [6 95]

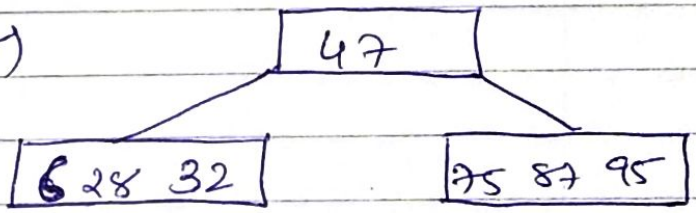
3) [6 47 95]

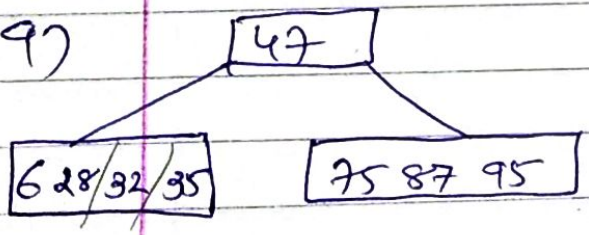
4) [6 32 | 47 | 95]

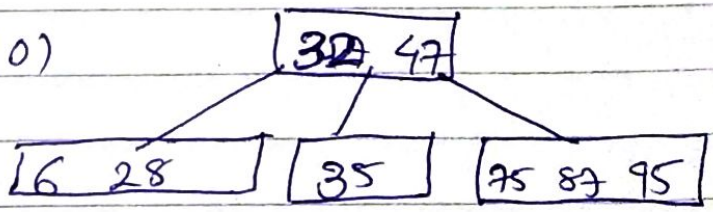
5)


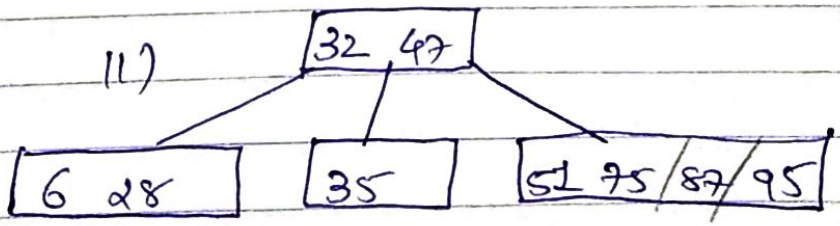
6)


7)


8)


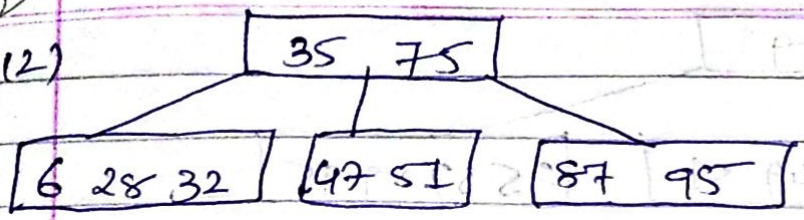
9)


10)


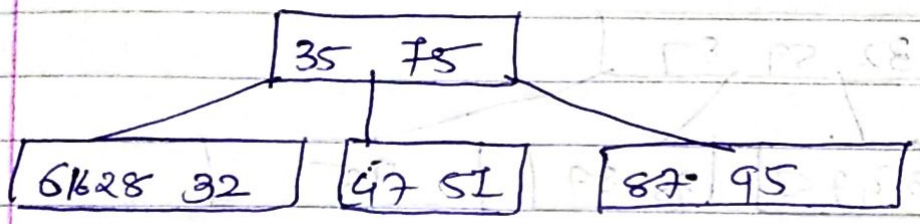
11)


12

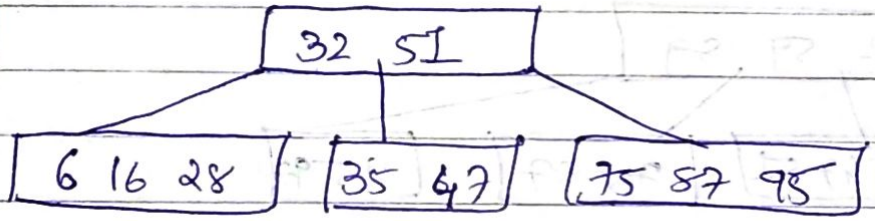
(12)



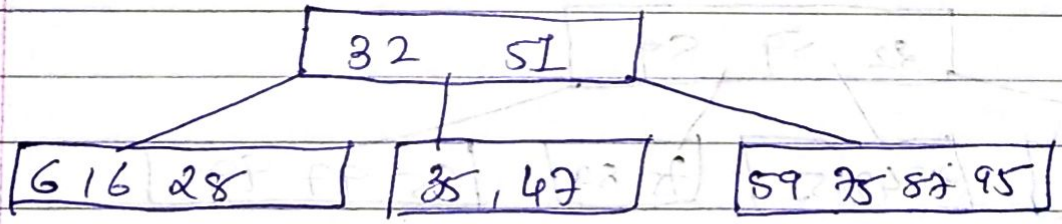
(13)



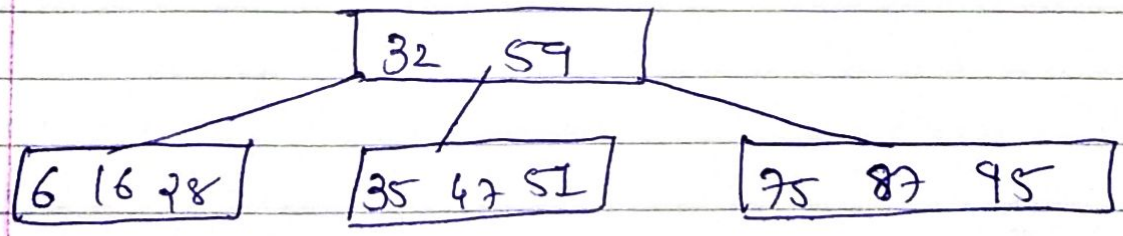
(14)



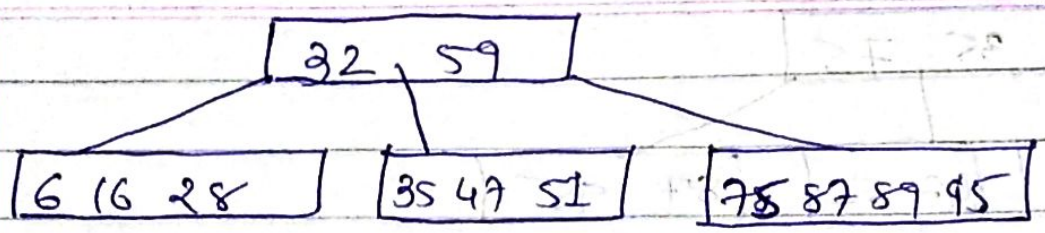
(15)



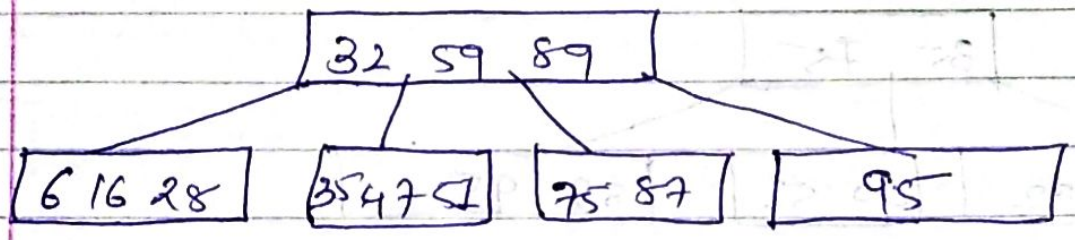
(16)



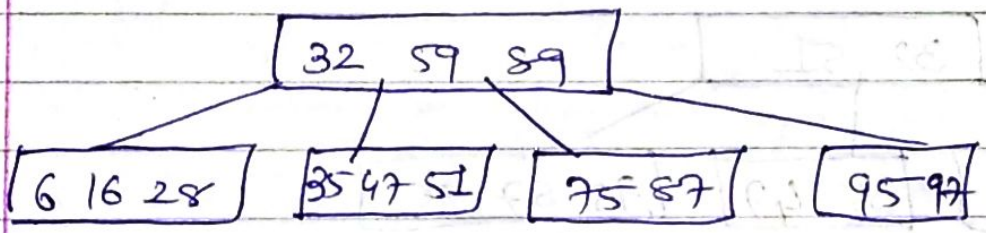
(7)


32 59
6 16 28
35 47 51
75 87 89 95

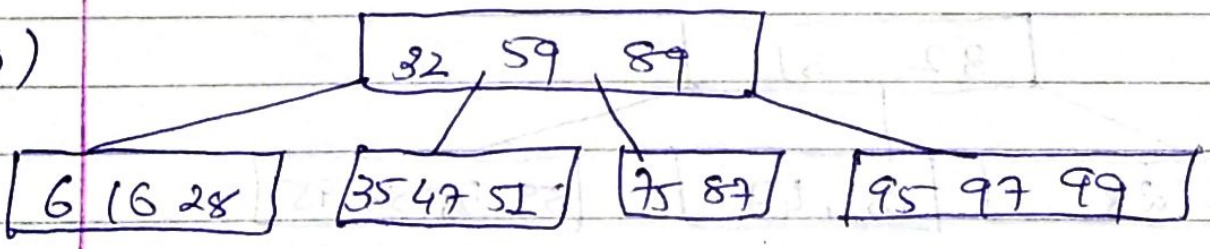
(8)


32 59 89
6 16 28
35 47 51
75 87
95

(9)


32 59 89
6 16 28
35 47 51
75 87
95 97

(20)


32 59 89
6 16 28
35 47 51
75 87
95 97 99