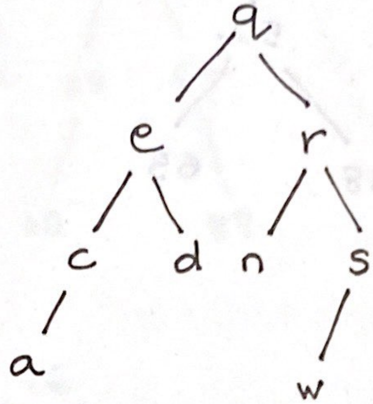
ASSIGNMENT 4 PRD190001 · CS5343.001

Q1)




a) inorder traversal:
acedqnrws

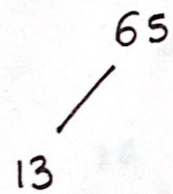
b) preorder traversal
qecadrnsw

c) postorder traversal
acdenwsrq

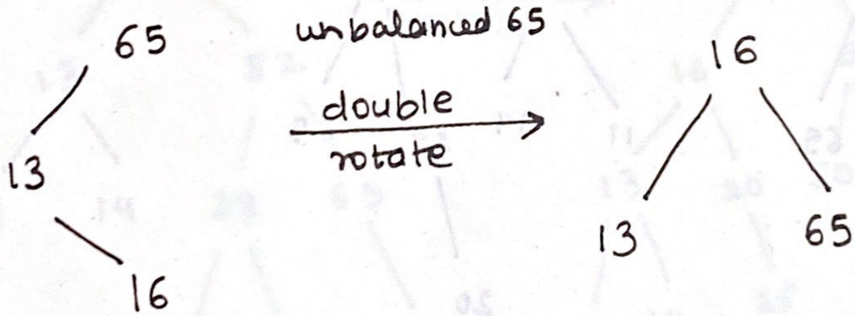
Q2)

1) 65

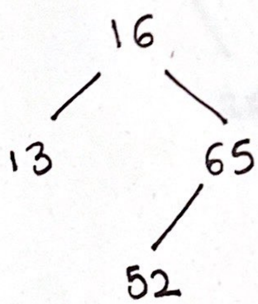
2)



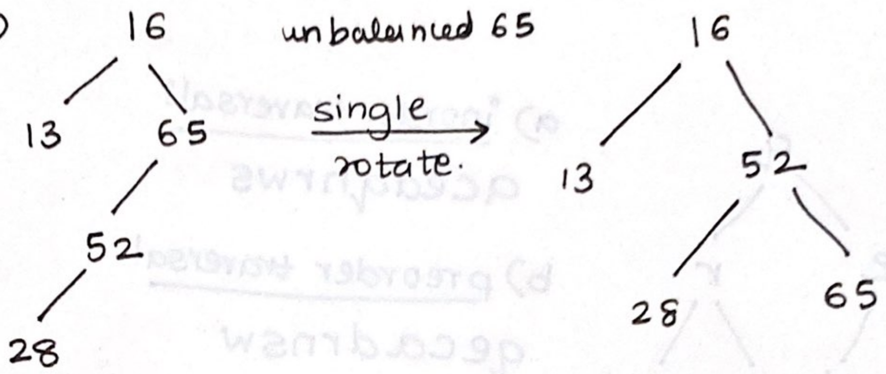

3)



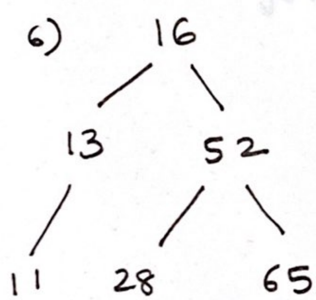

4)



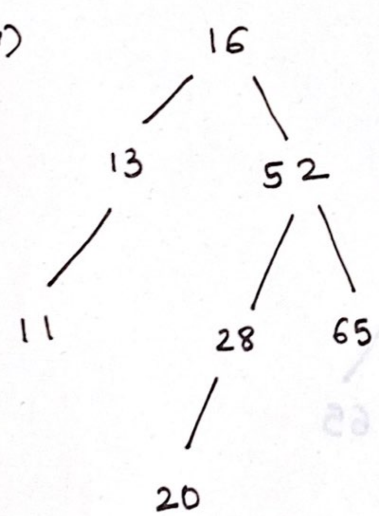
5)


16
13
65
52
28
unbalanced 65
single
rotate.
16
13
52
28
65

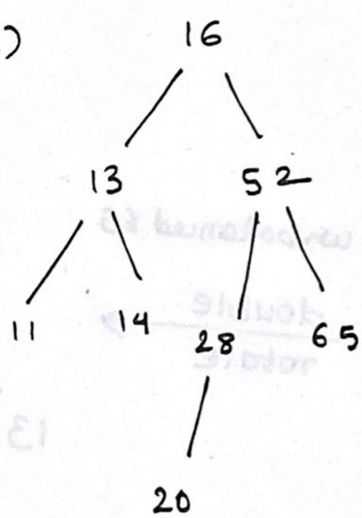
6)


16
13
52
11
28
65

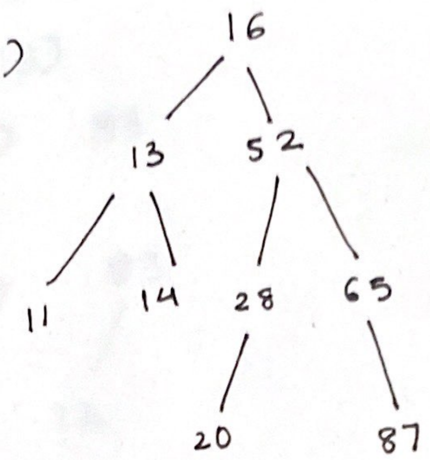
7)


16
13
52
11
28
65
20

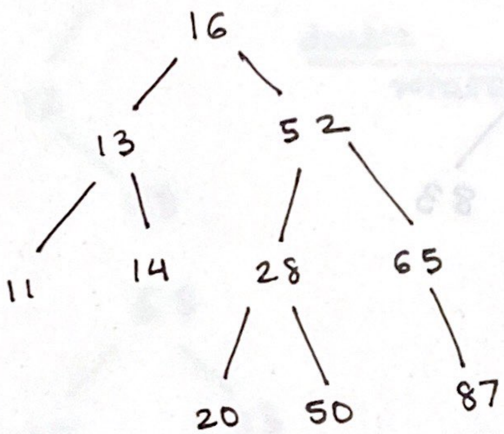
8)


16
13
52
11
14
28
65
20

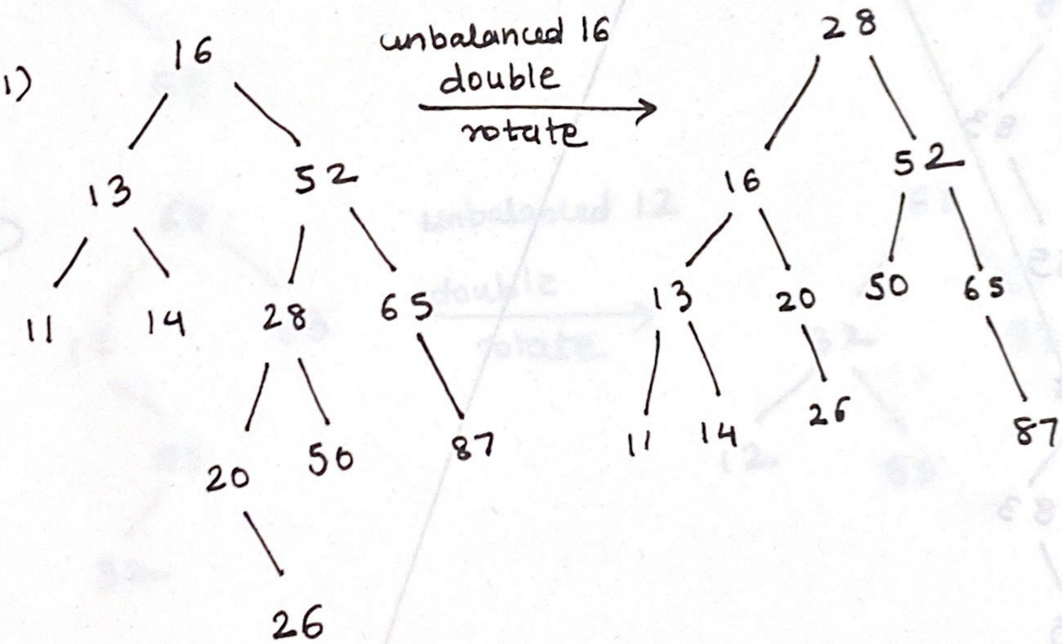
9)


16
13
52
11
14
28
65
20
87

10)


16
13
52
11
14
28
65
20
50
87

11)

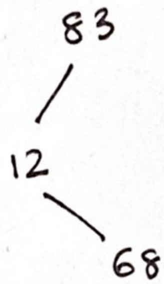

16
13
52
11
14
28
65
20
50
87
26
unbalanced 16
double
rotate
28
16
52
13
20
50
65
11
14
26
87

Q3)

1) 83

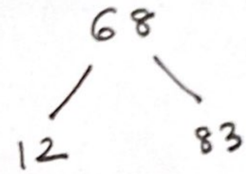
2)


3)

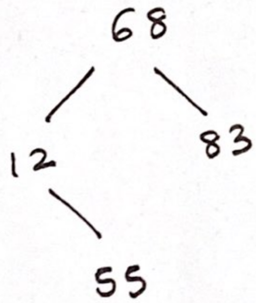



unbalanced 83

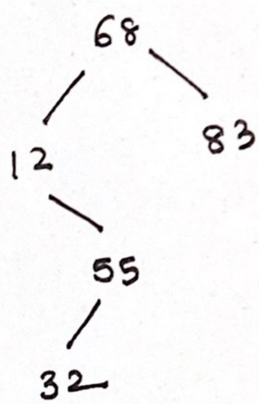
double rotate →




4)

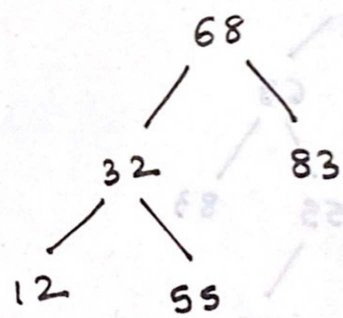



5)

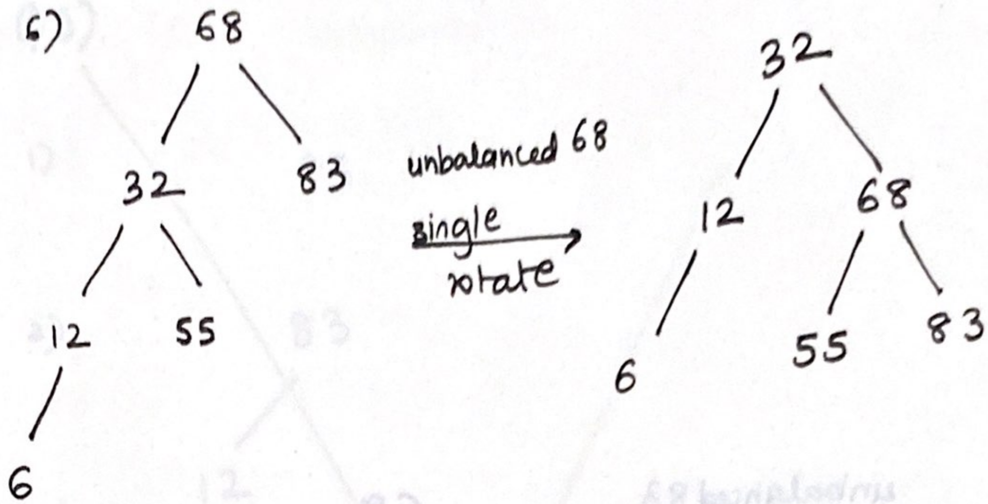



unbalanced 12

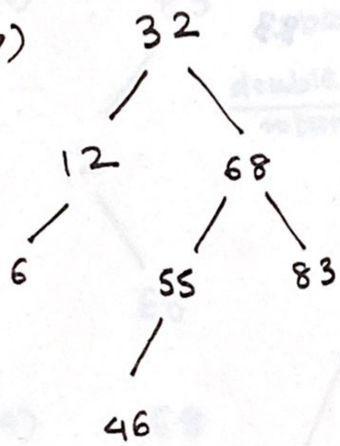
double rotate →



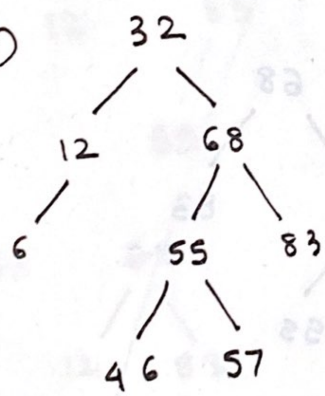
6)


68
32
83
12
55
6
unbalanced 68
single rotate
32
12
68
6
55
83

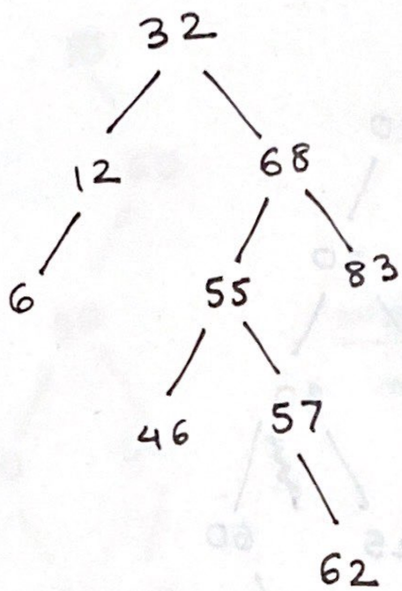
7)


32
12
68
6
55
83
46

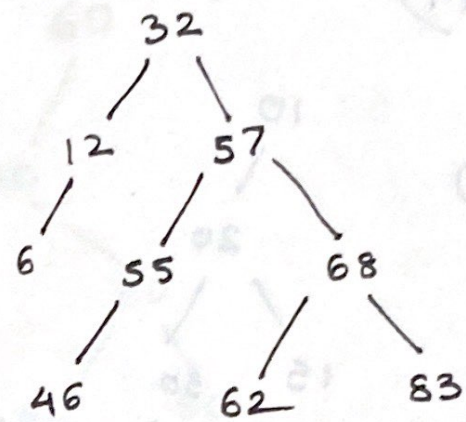
8)


32
12
68
6
55
83
46
57

9)

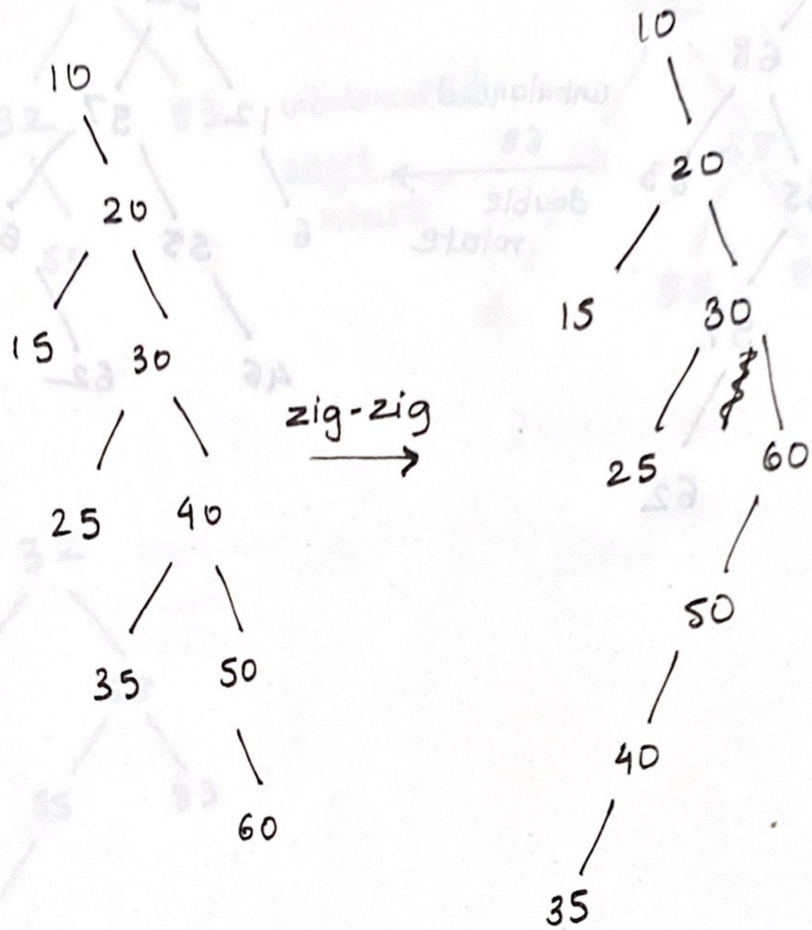



unbalanced 68 → double rotate

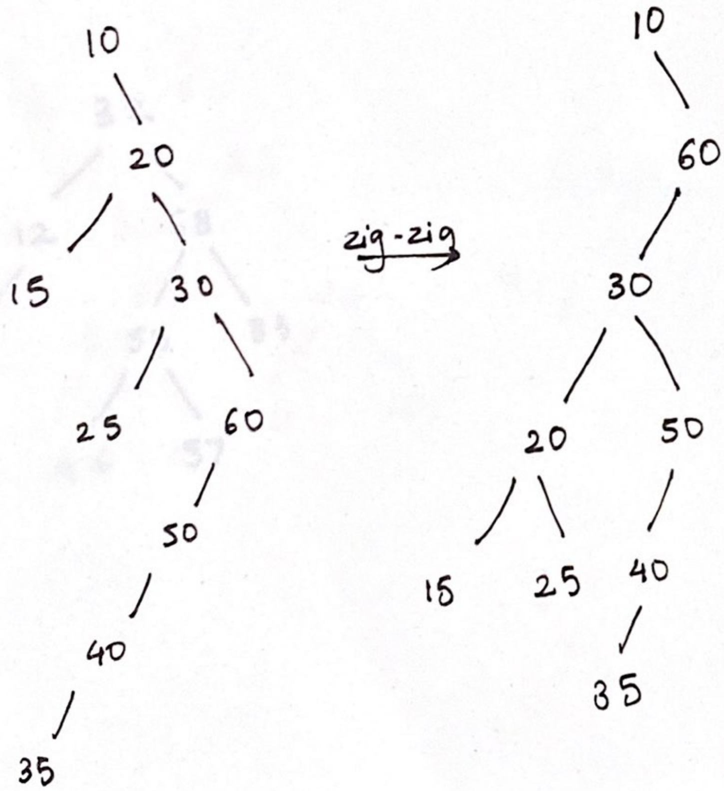


Q4)

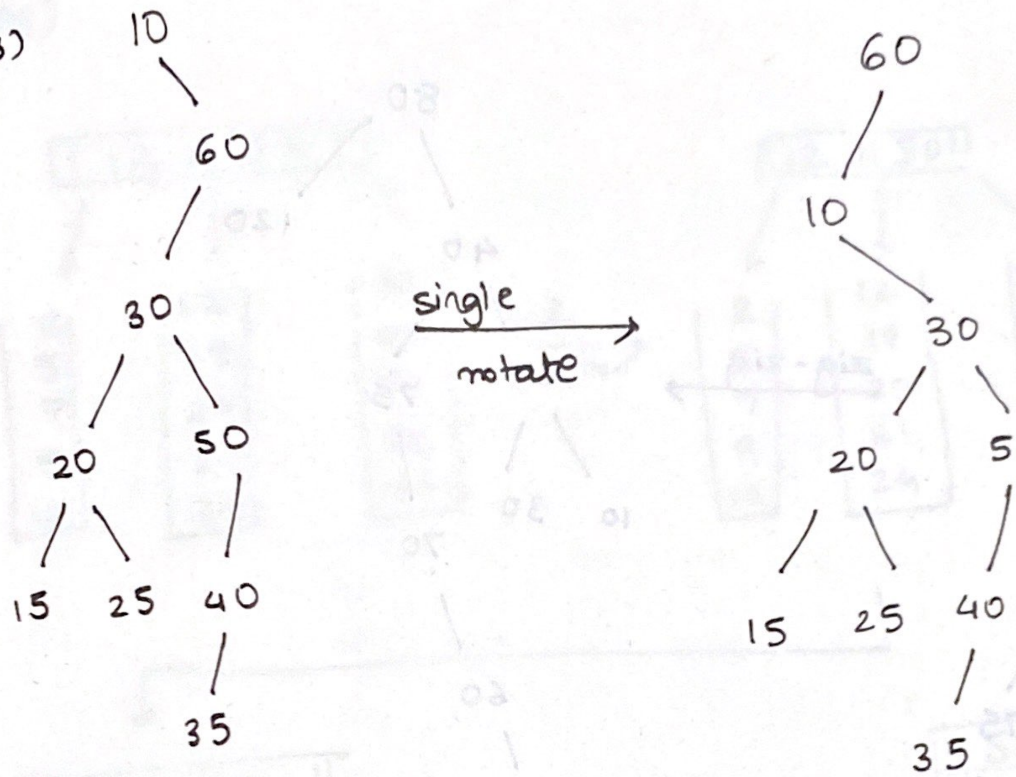
1)


10
20
15
30
25
40
35
50
60
zig-zig
10
20
15
30
25
60
50
40
35

2)

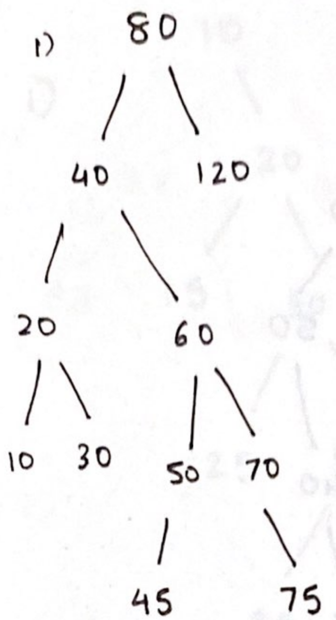

10
20
15
30
25
60
50
40
35
zig-zig
10
60
30
20
50
15
25
40
35

3)

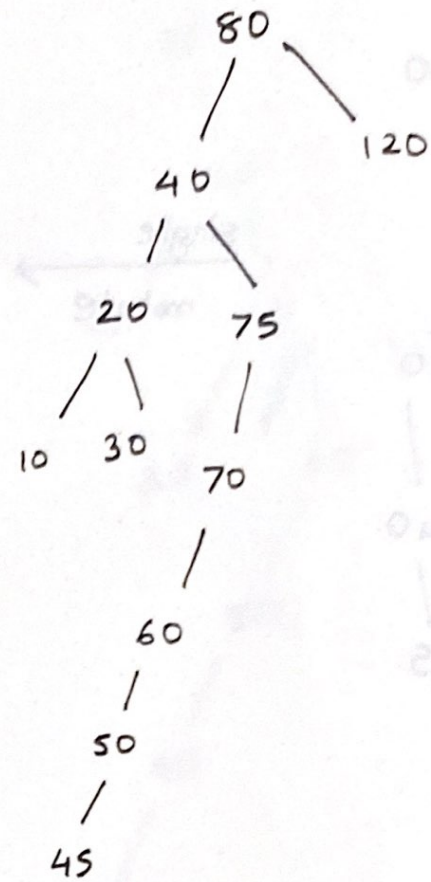


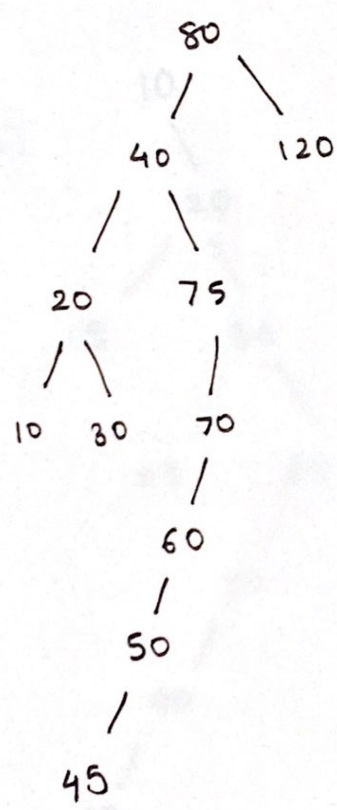
Q5)

1)

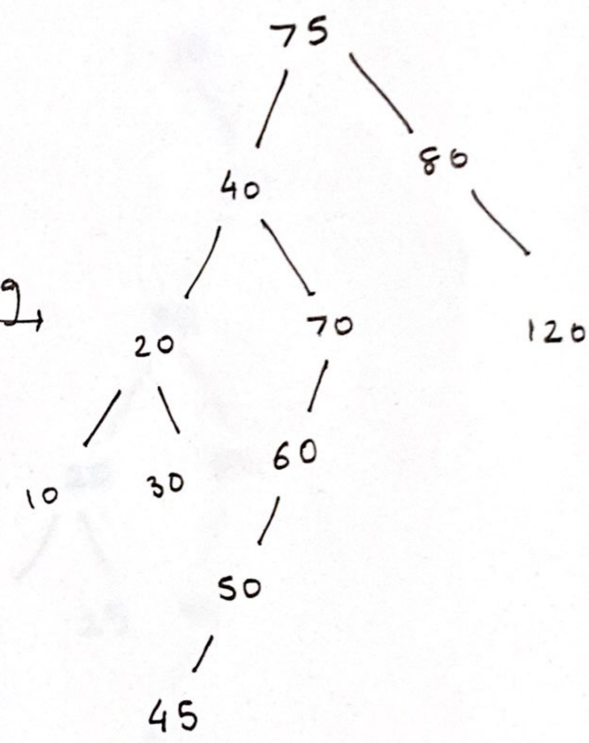



zig-zig →

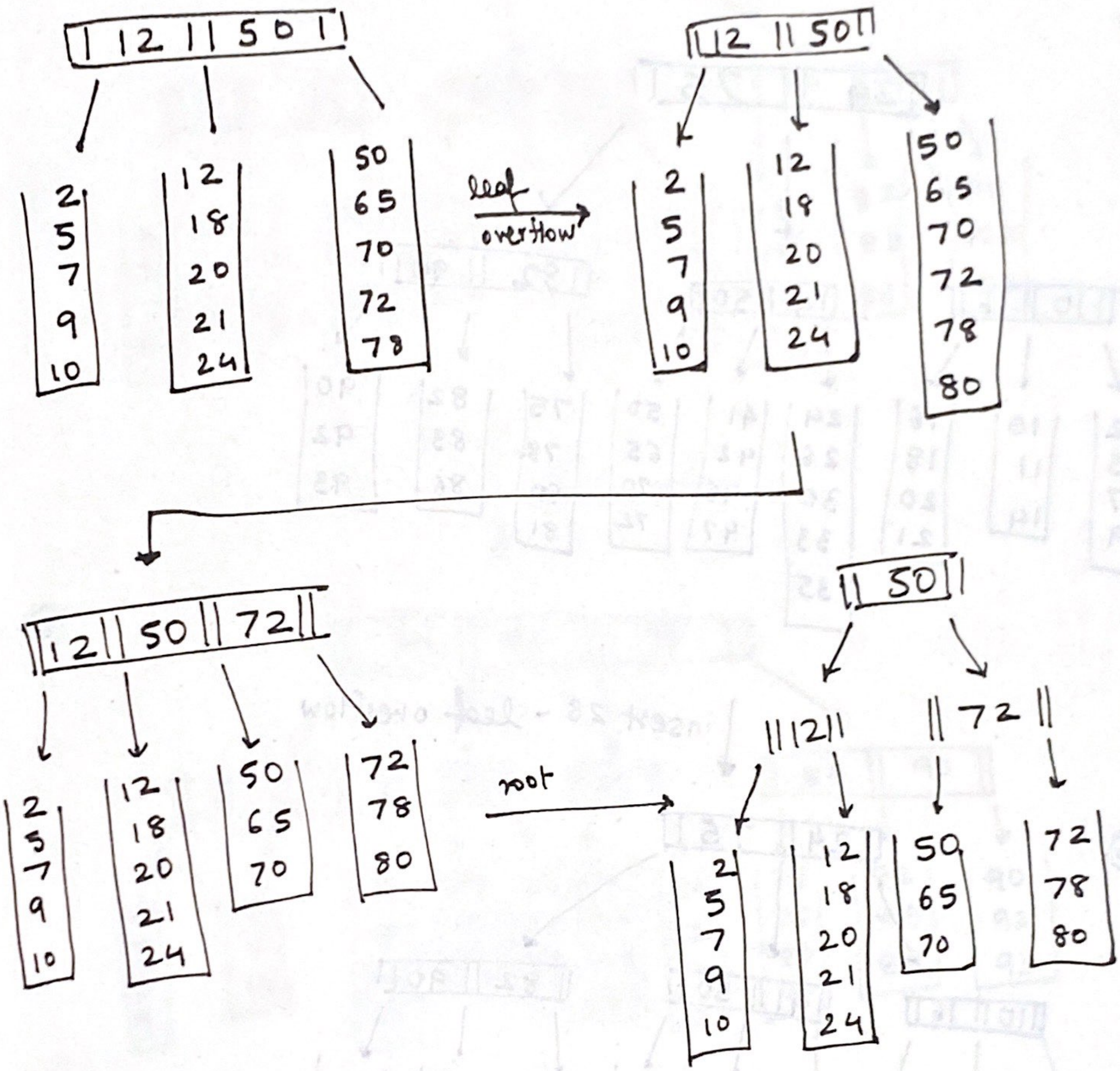



zig-zag →



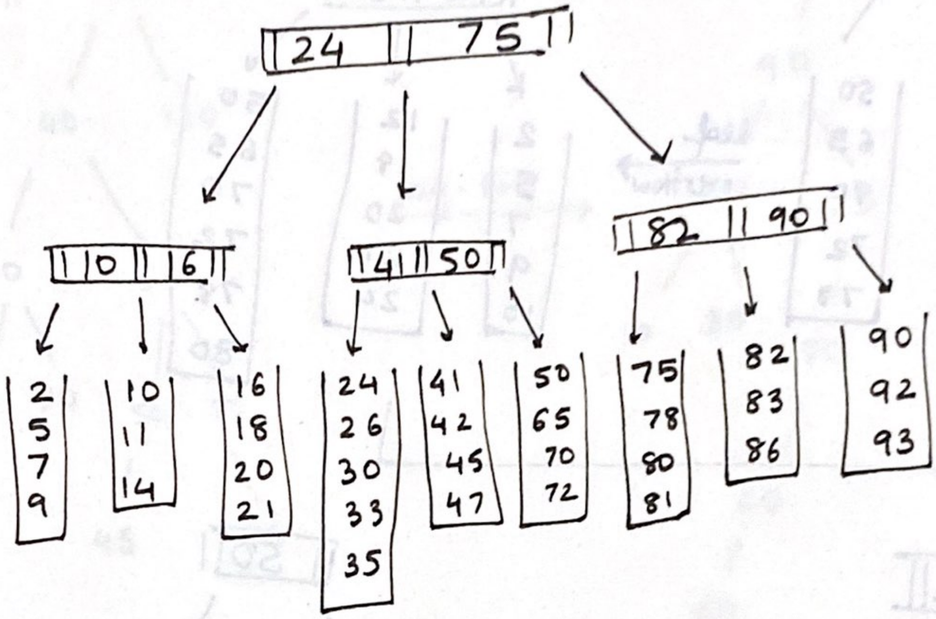
Q6)



12 50
2 5 7 9 10
12 18 20 21 24
50 65 70 72 78
leaf overflow
12 50
2 5 7 9 10
12 18 20 21 24
50 65 70 72 78 80
12 50 72
2 5 7 9 10
12 18 20 21 24
50 65 70
72 78 80
root
50
12
72
2 5 7 9 10
12 18 20 21 24
50 65 70
72 78 80

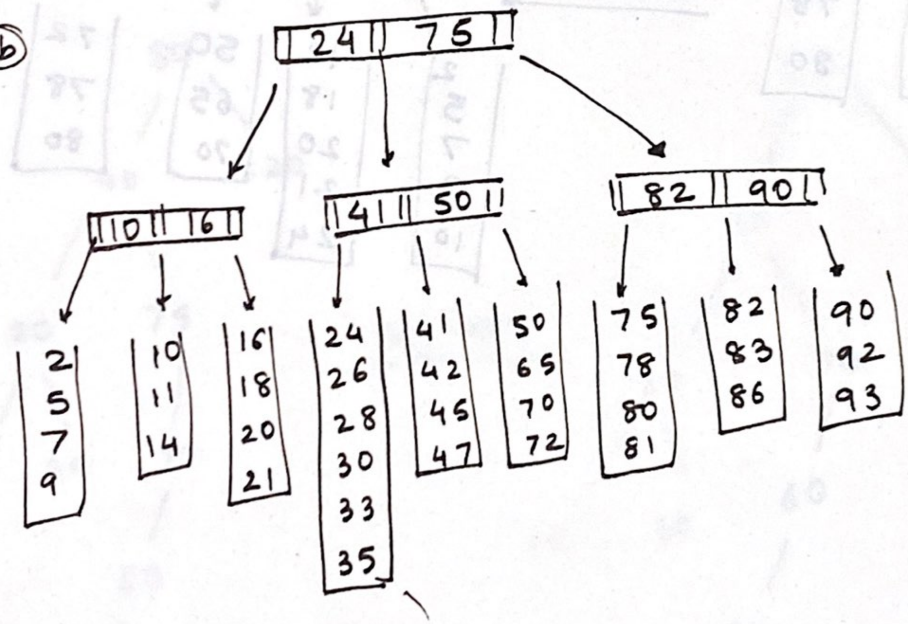
Q7)

(a)

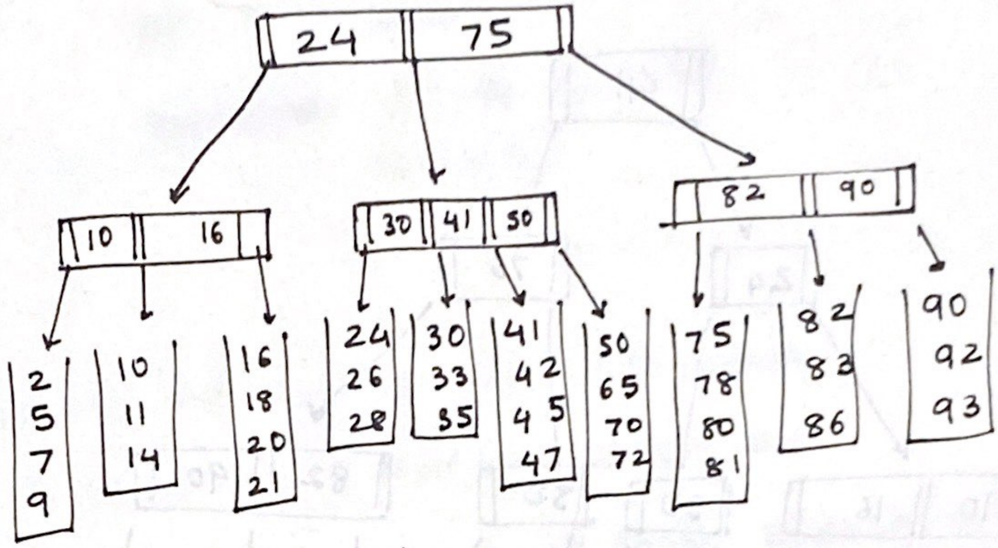


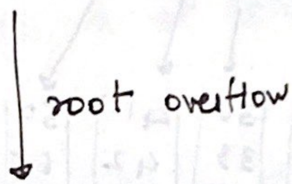

insert 28 - leaf overflow

(b)

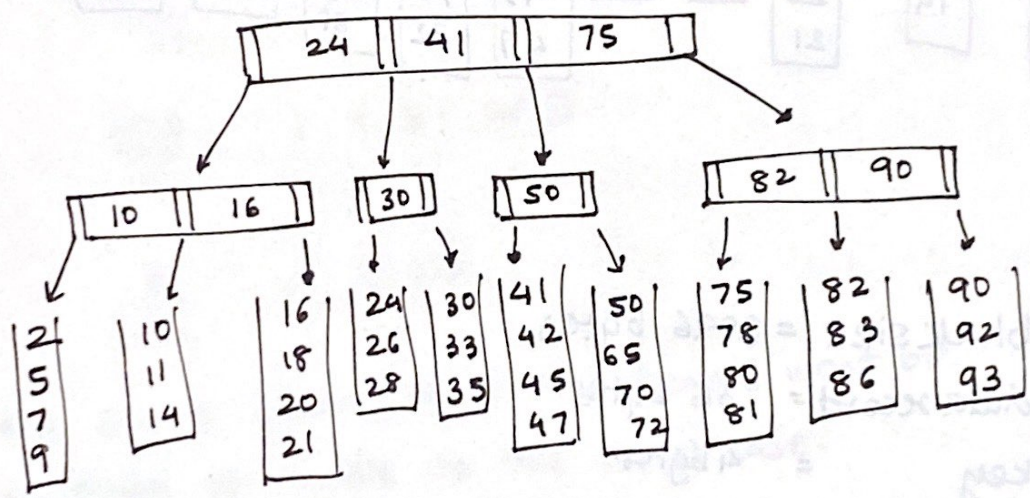


(c)

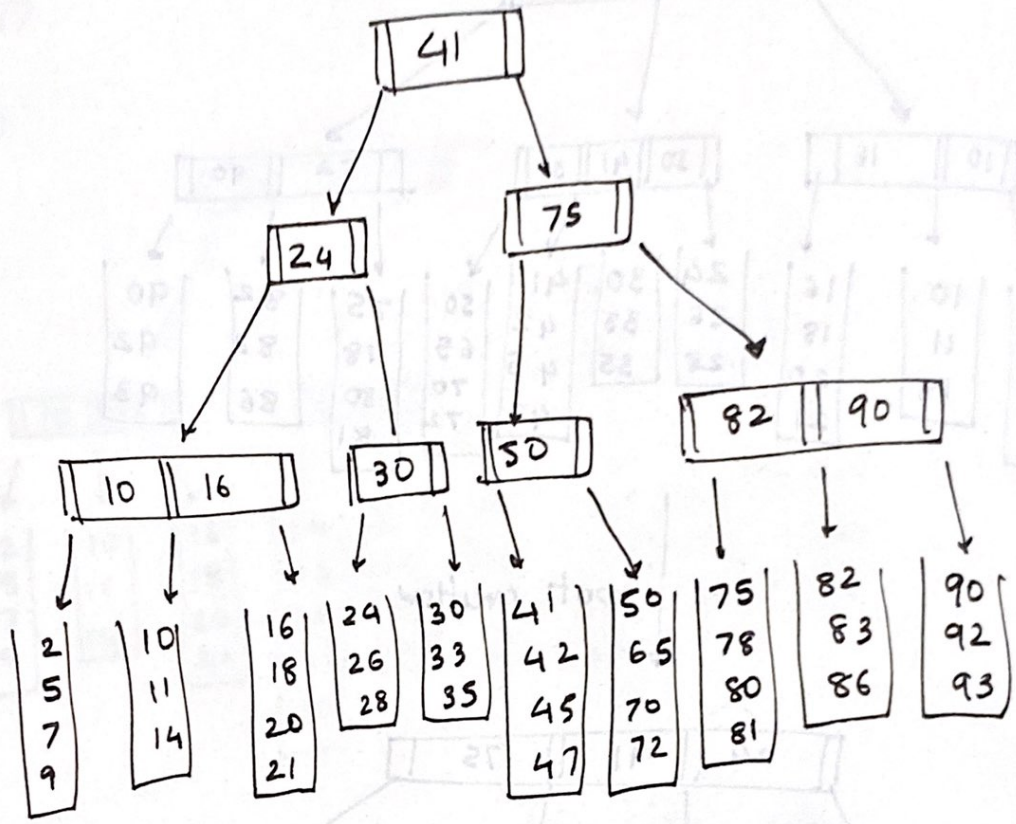





(d)



(e)




Q8)

block size = 3096 bytes

data record = 36 bytes

key = 4 bytes.

pointer = 4 bytes.

$$L = \frac{\text{blocksize}}{\text{data record}} = \frac{3096}{36} = \underline{\underline{86}}.$$

Total for one node → M-1 keys

M pointers.

$\therefore 4(M-1) + 4M = 8M - 4$

$$8M - 4 \le 3096$$

$$8M \cancel{-4} \le 3100$$

$$M = 387$$

Q9)

Since each leaf could be half of full capacity.

8,600,000 records could be in 200000 leaves.

if each internal node is at least branching $\lceil \frac{M}{2} \rceil$ times ie 194 then

$$\frac{200000}{194} = 1030.92 \approx 1031$$

$$\frac{1031}{194} = 5.31 \approx 5$$

$$\frac{5}{194} = 0.03$$

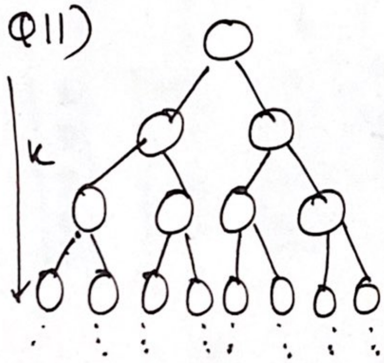
So the above shows that the leaves wont be deeper than 4 levels in the worst case.

Q10) A binary tree with N nodes will have N-1 pointers.

but as each node can have 2 children the total number of pointers = 2N.

$$\therefore \quad 2N - (N-1) = 2N - N + 1$$
$$= \underline{\underline{N+1}} \text{ null pointers}$$

Q11)




A binary tree has $2^{k-1}$ leaves.

As we see that the binary tree is perfect, filled to all the levels.

Lets say the above tree has n nodes. out of which $2^{k-1}$ are leaf nodes. Where k is the depth

If we add another level we need to add ($2^{k-1} * 2$) nodes to the tree which will make leaf nodes from $\underline{2^{k-1} \text{ to } 2^k}$

& the nodes will be

$$\underline{(n) + (n+1) = 2n+1}$$