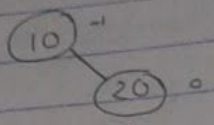
AVL Trees:

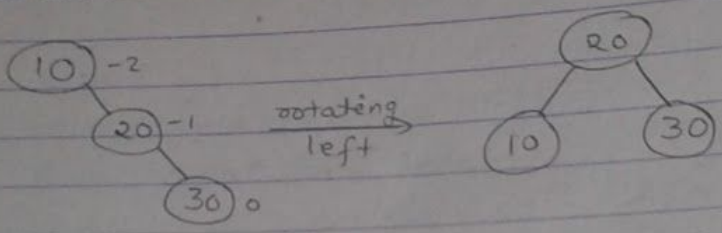
① Create a AVL balanced tree for data sequence
10, 20, 30, 50, 45, 40, 8, 5, 3

Step1: Insert 10

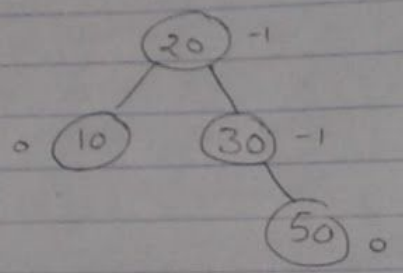
Step2: Insert 20



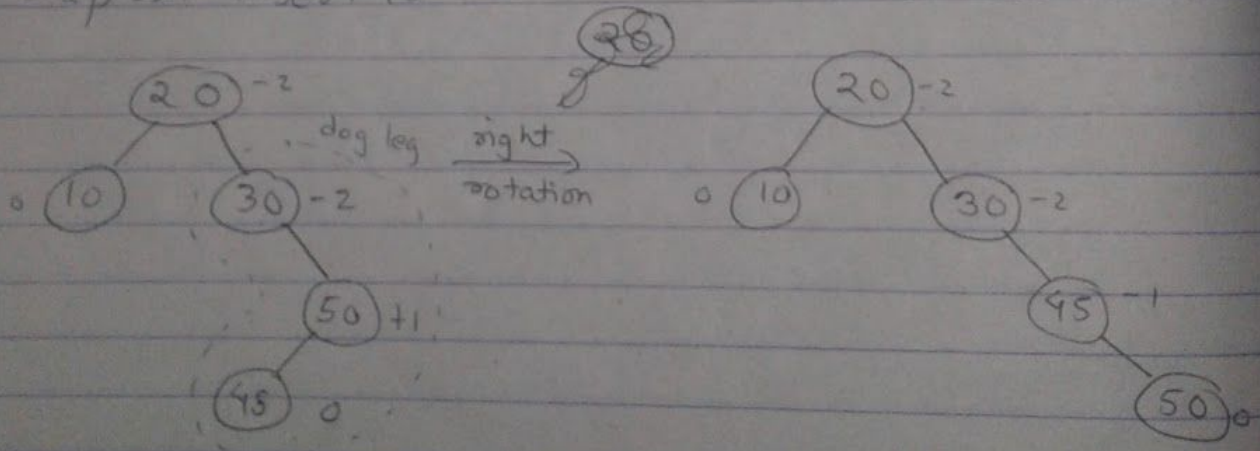
Step3: Insert 30



Step4: Insert 50

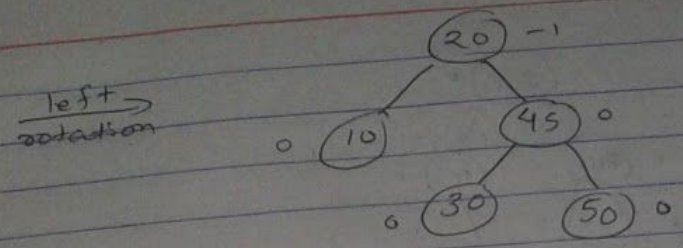


Step 65: Insert 45

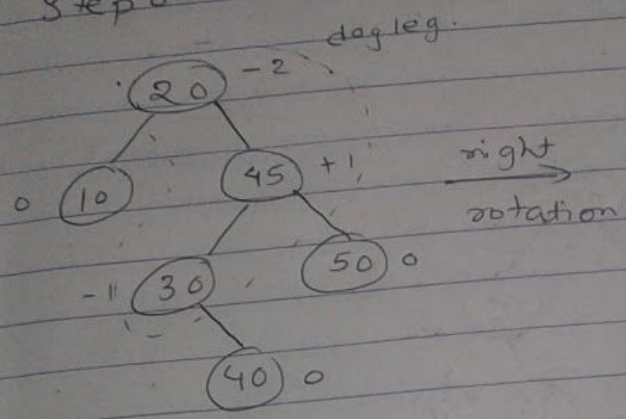


temp;
temp = table[rem];
while(temp->next != NULL)
temp = temp->next;
temp->next = p;

3) Draw the hash functions :- ,

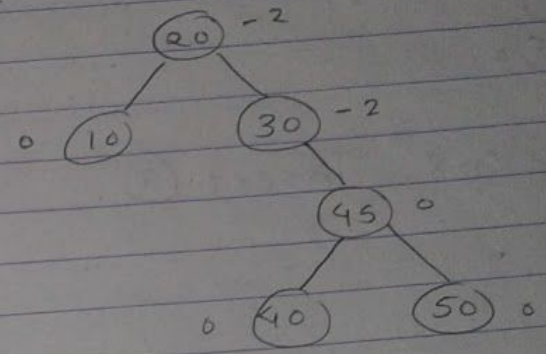


Step 6: Insert 40

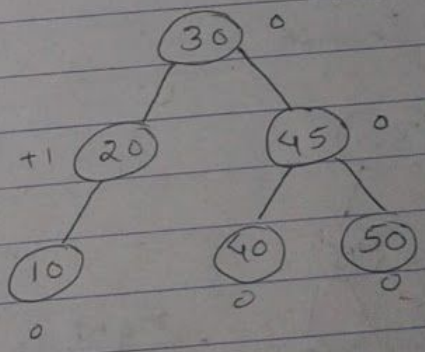


right rotation

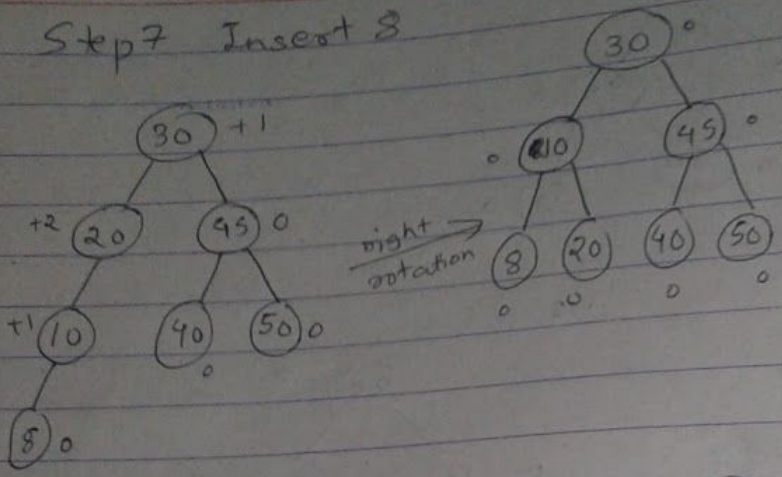
2



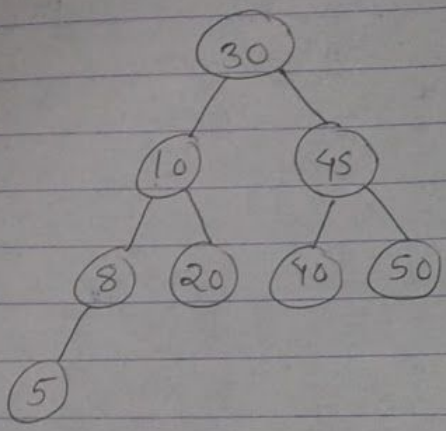
left rotation



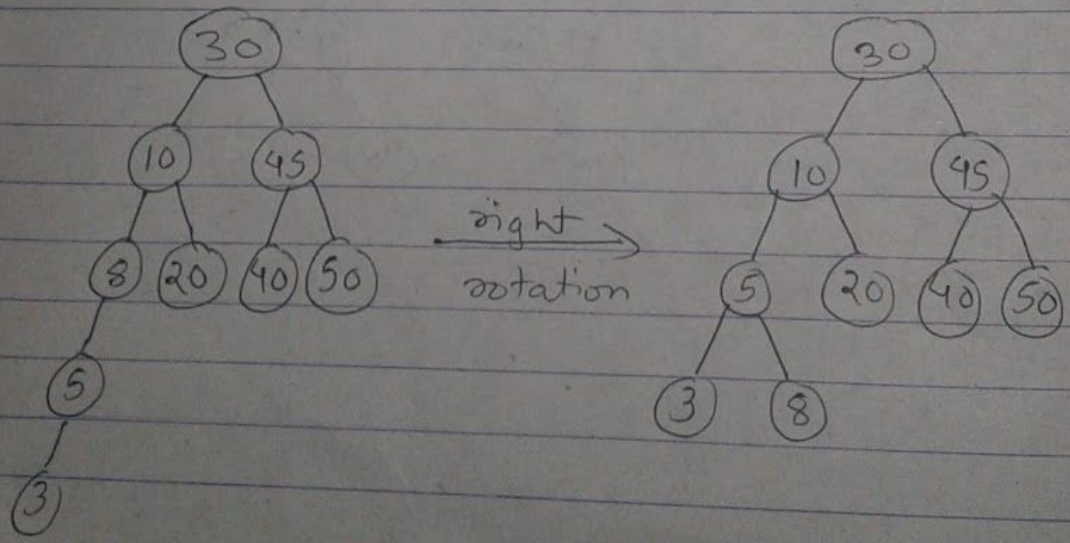
Step 7 Insert 8



Step 8. Insert (5)



Step 9. Insert 3



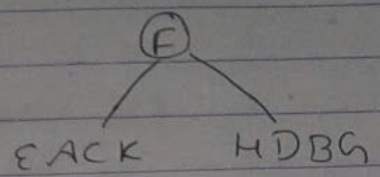
17000

3) Draw the hash functions :- 1
keys.

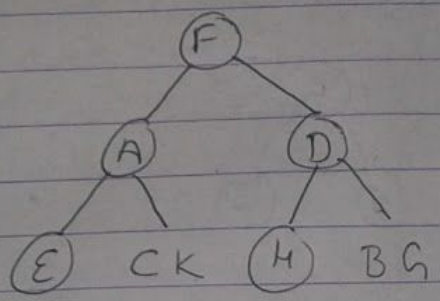
(temp = temp->next;
temp->next != NULL)
temp->next = p;

(2) Draw a binary tree:
Preorder FAEKCDHGB
Inorder EACKFHDBG

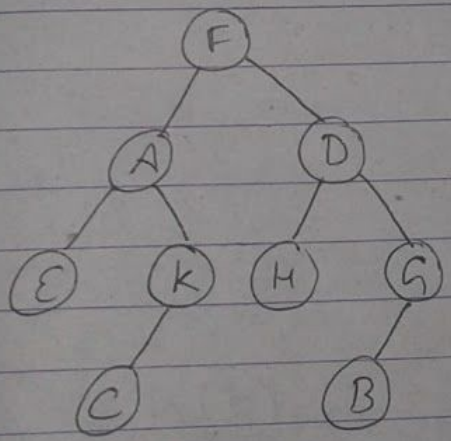
Step1



Step2



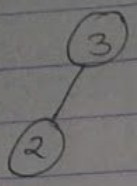
Step3



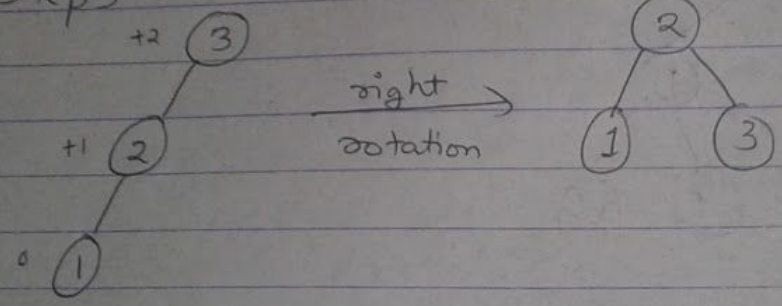
(3) Construction AVL tree for:
3, 2, 1, 4, 5, 6, 7, 16, 15, 14, 13, 12

Step 1

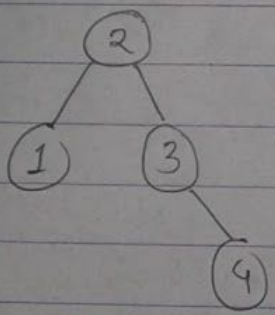
Step 2



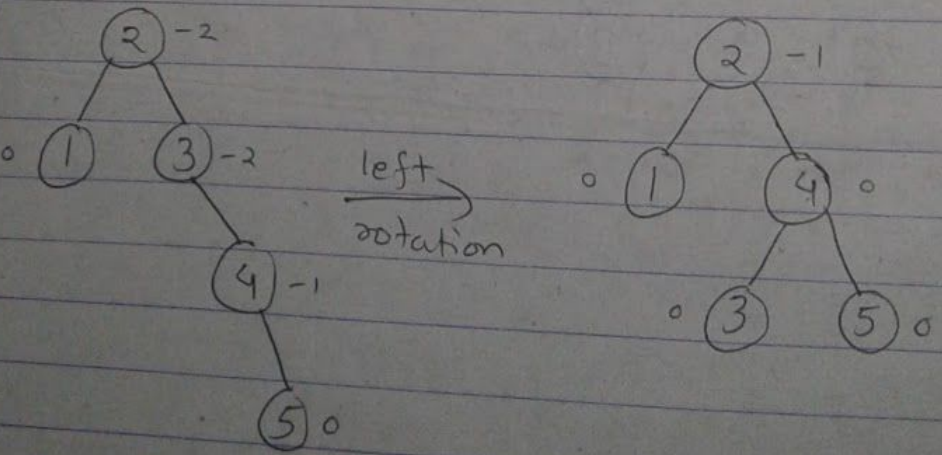
Step 3

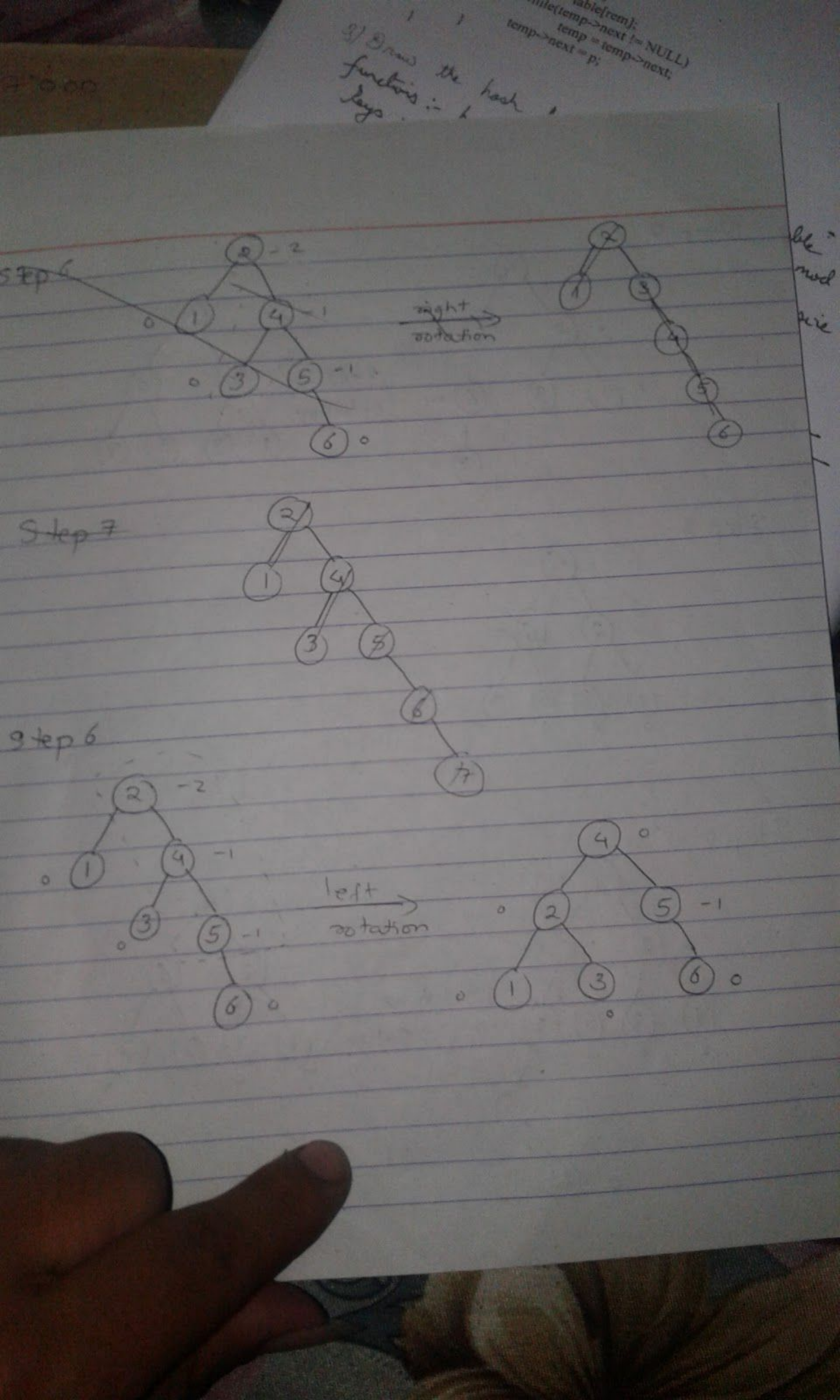


Step 4

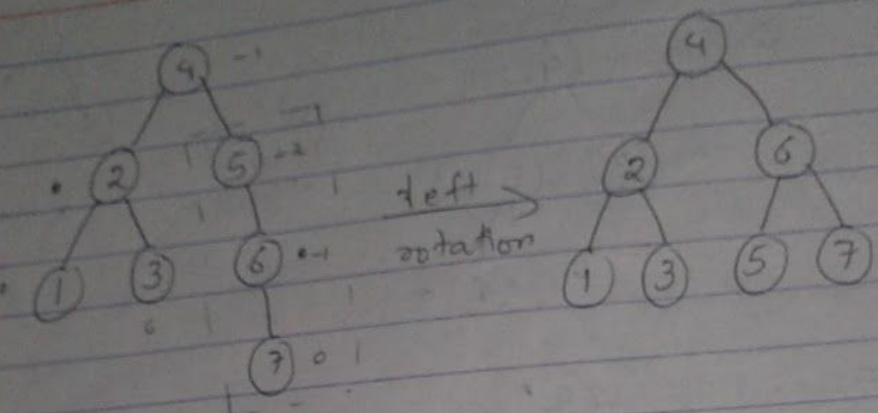


Step 5.

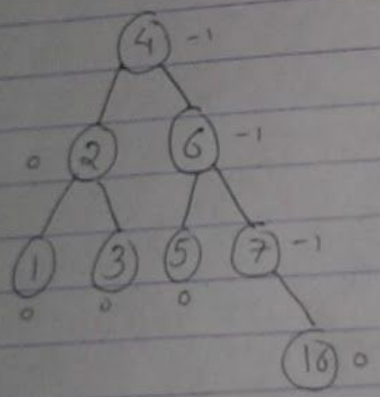



Step 6
right rotation
Step 7
step 6
left rotation

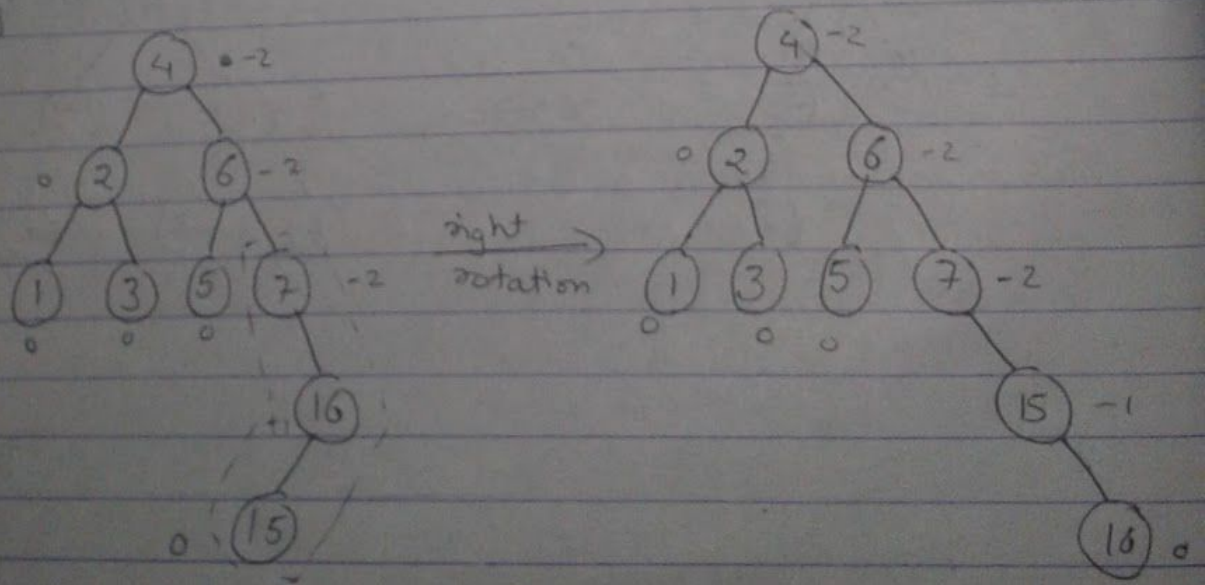
Step 7

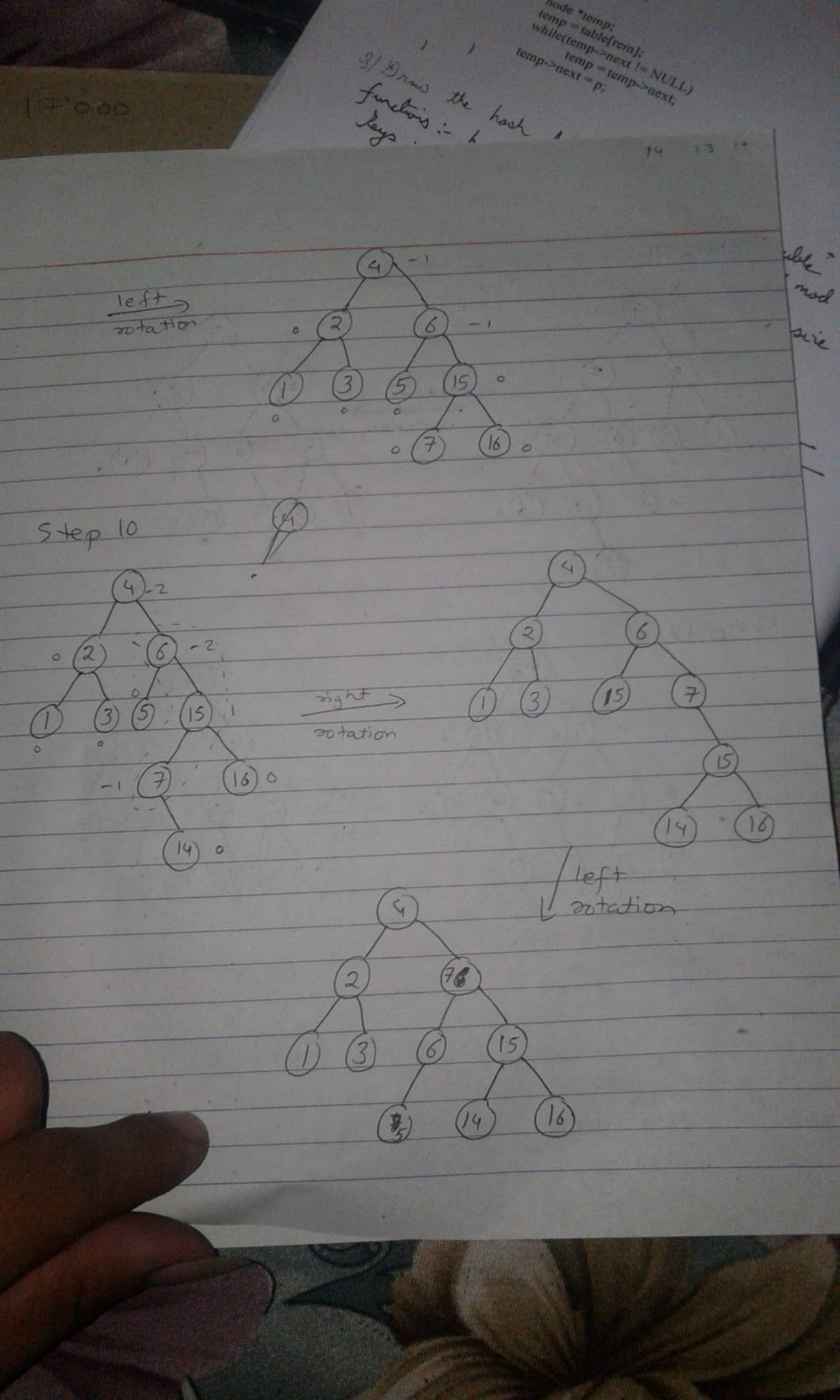


Step 8

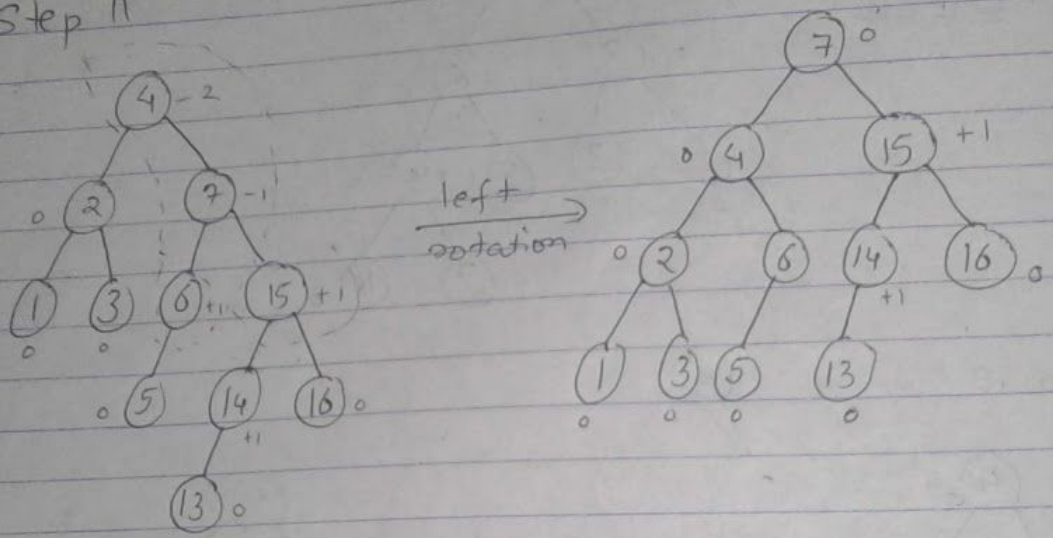


Step 9

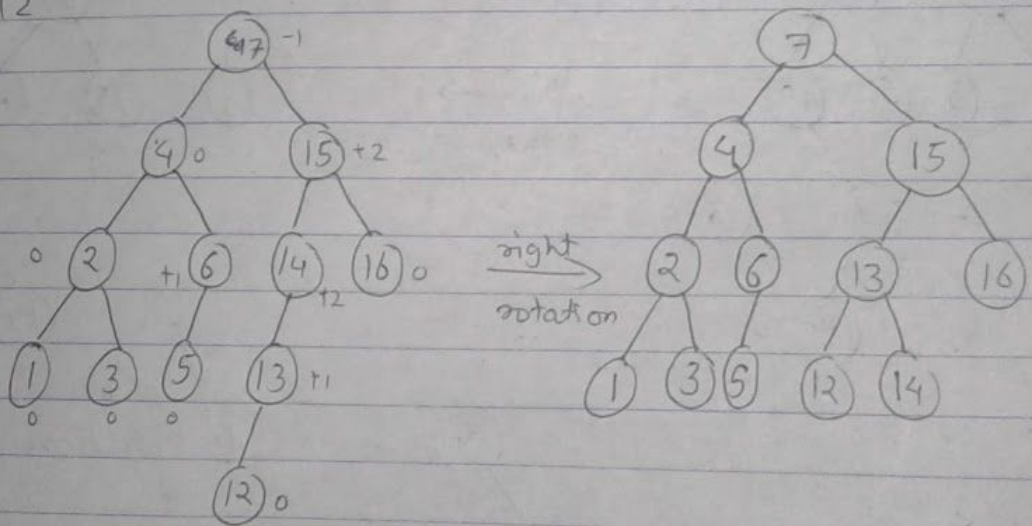



left rotation
Step 10
right rotation
left rotation

Step 11



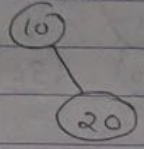
Step 12



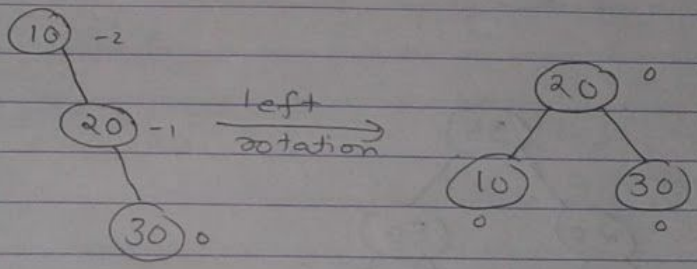
4) Create an AVL balanced tree for the set of data 10, 20, 30, 35, 50, 70, 40, 80, 60, 65.

Step 1

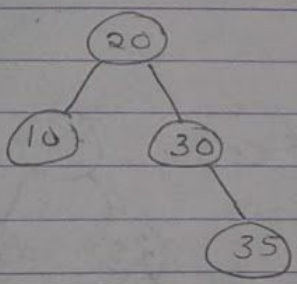
Step 2



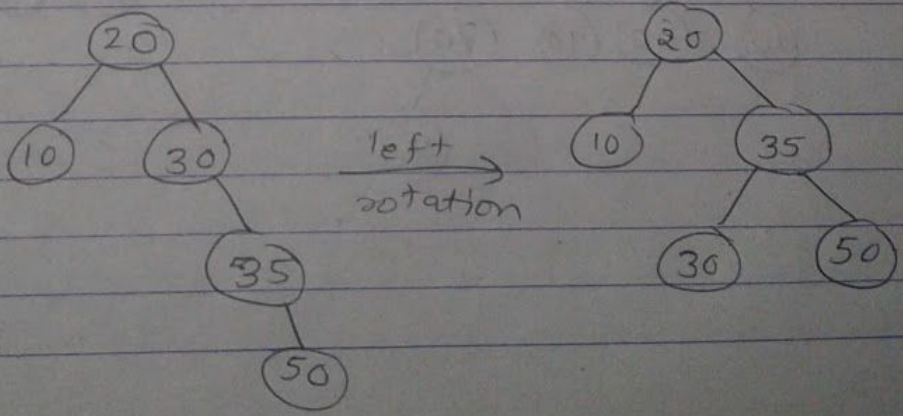
step 3



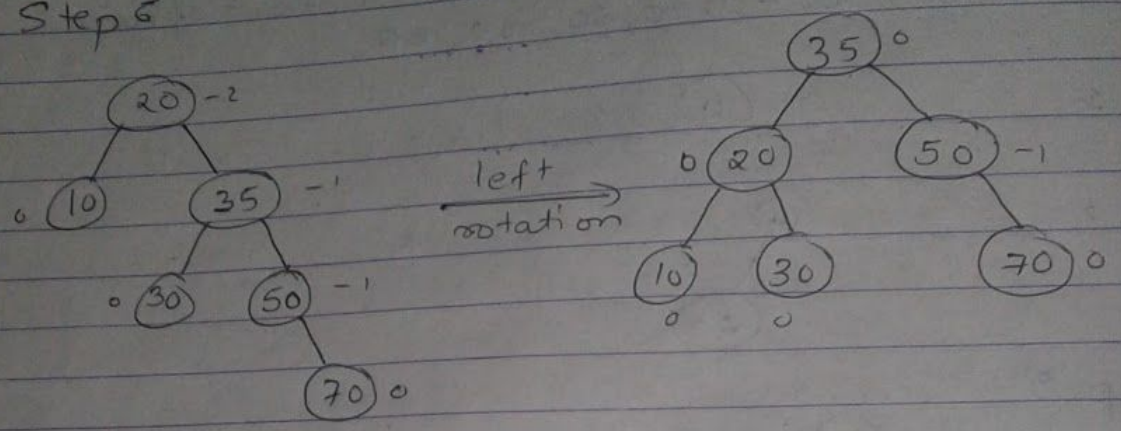
Step 4



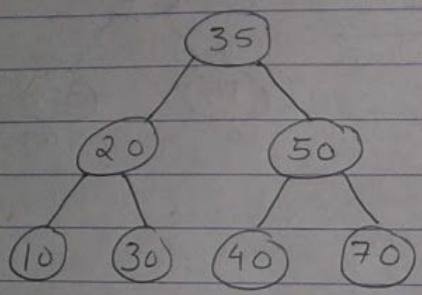
Step 5



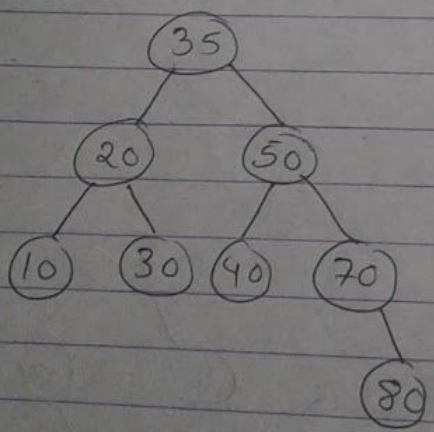
Step 6



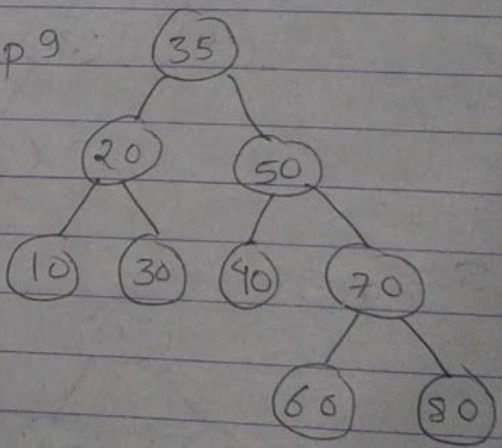
Step 7



Step 8

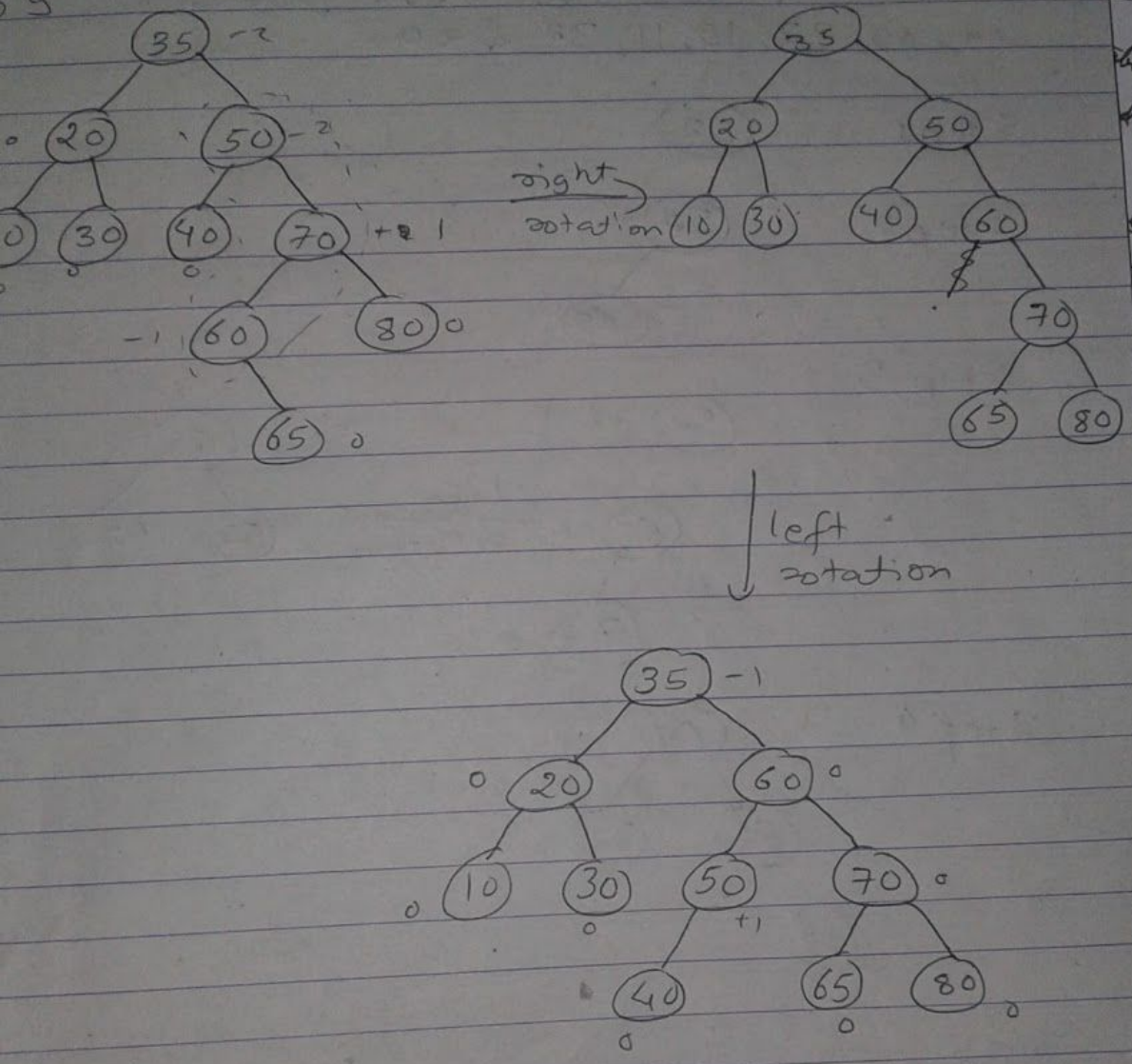


Step 9



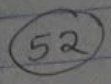
temp;
temp = table[rem];
while(temp->next != NULL)
temp = temp->next;
temp->next = p;

3) Draw the hash
functions :- h, (x, table
keys :- 7
us

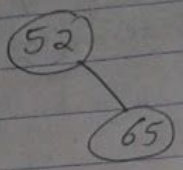


5) Construct a ~~recursion~~ AVL Tree for 52, 65, 72, 15, 11, 32 & 20

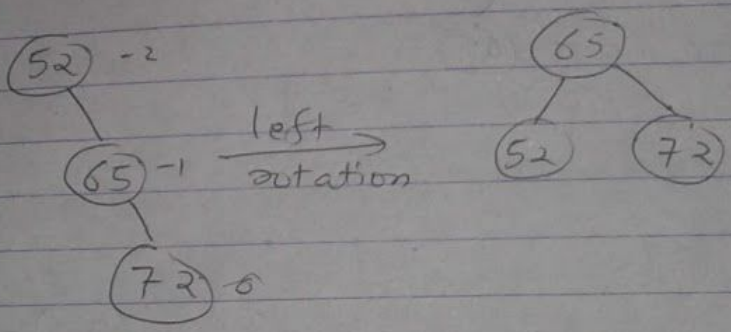
Step 1



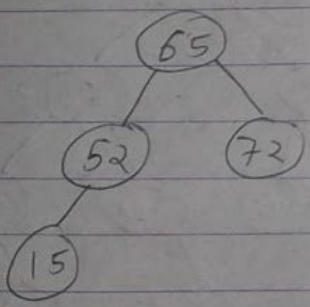
Step 2



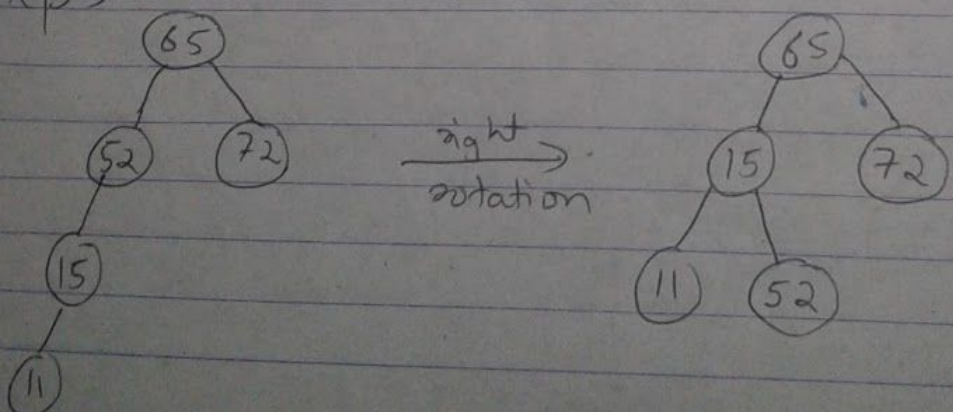
Step 3

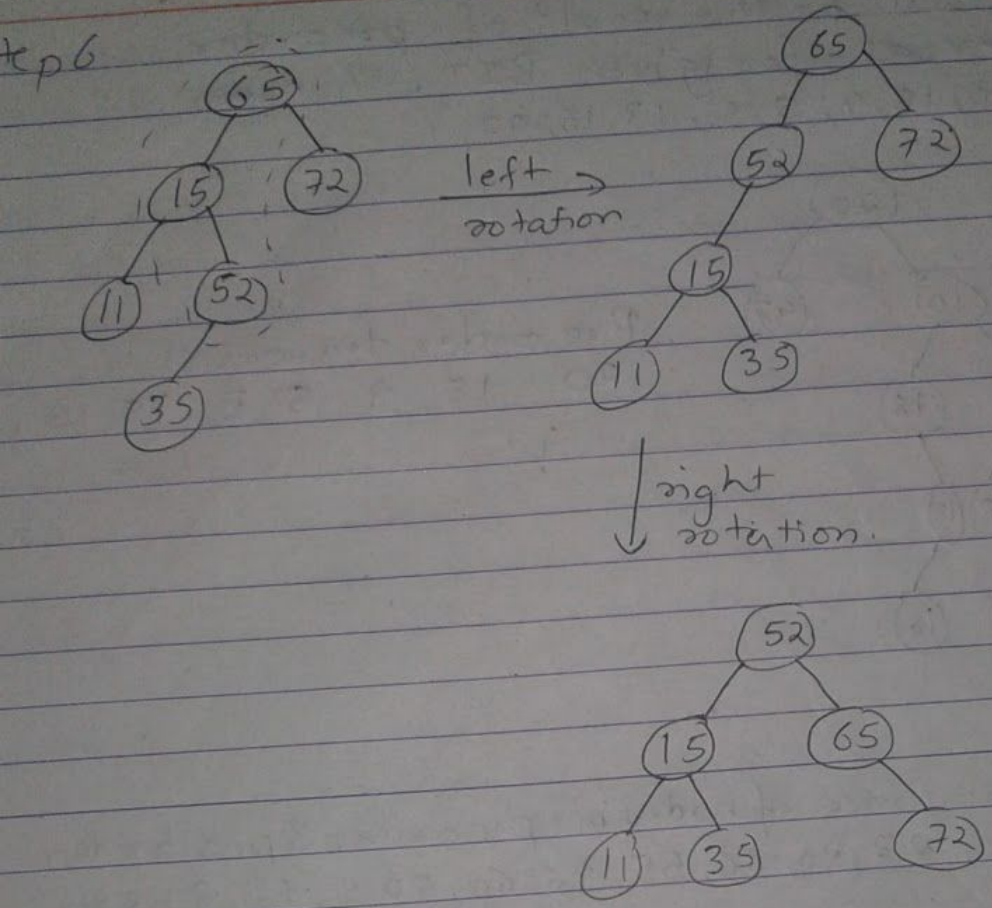


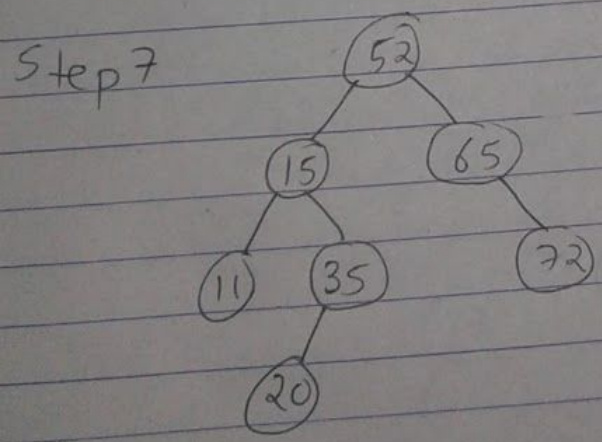
Step 4



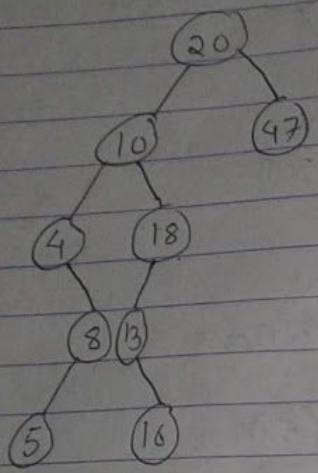
Step 5




tep6
65
15
72
11
52
35
left
rotation
65
52
72
15
11
35
right
rotation.
52
15
65
11
35
72

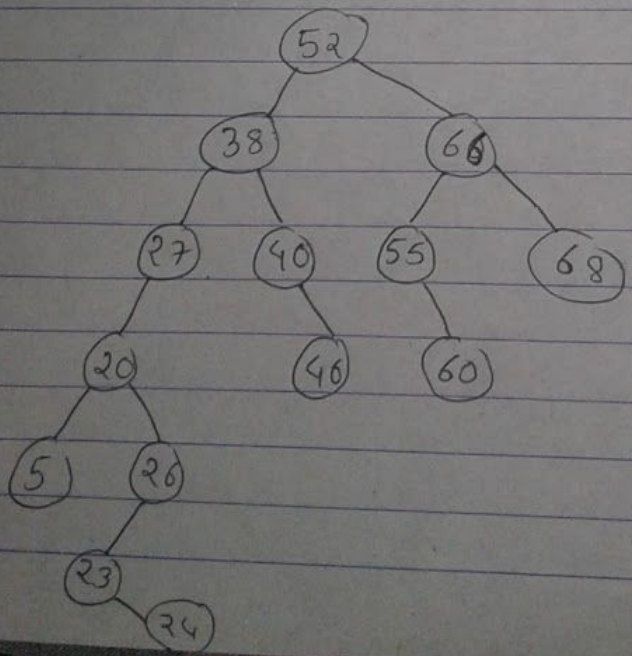

Step 7
52
15
65
11
35
72
20

6) What will be the result of preorder traversal for given BST with nodes
20, 10, 18, 4, 8, 5, 13, 16, 47,



Pre order traversal
20 10 4 8 5 18 13 16
47

7) Show the sequence of nodes in preorder & post order
52, 38, 40, 27, 20, 66, 55, 46, 60, 26, 5, 68, 23, 24



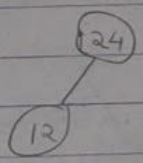
Preorder
52 38 27 20 5 26 23 24 40 46 66 55 60 68

Post order
5 24 23 26 20 27 46 40 38 60 55 68 66 52

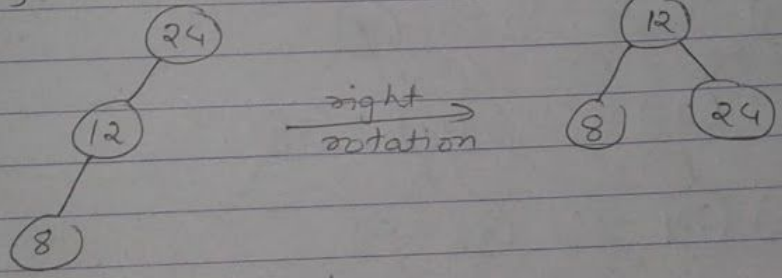
(8) Create an AVL tree from the data
24 12 8 15, 35 30 57 40 45 78

Step 1

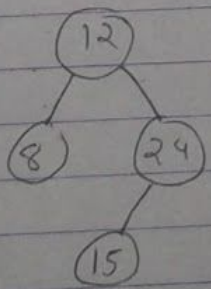
Step 2



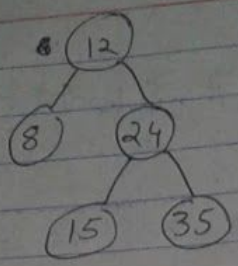
Step 3



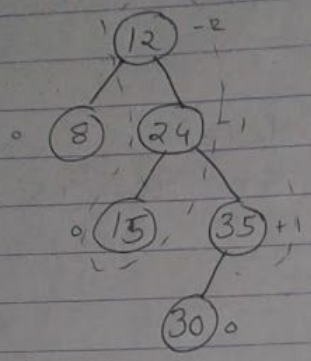
Step 4



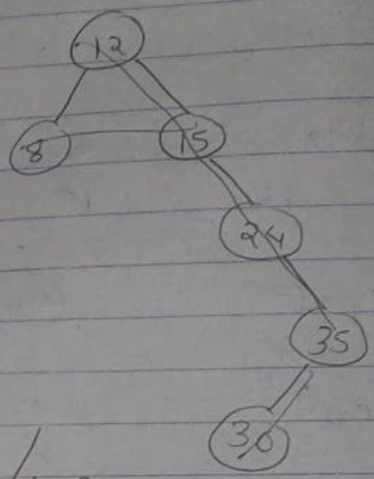
Step 5



Step 6

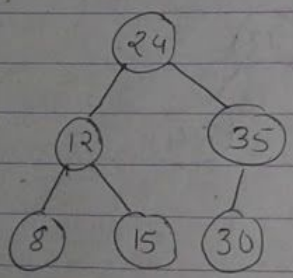


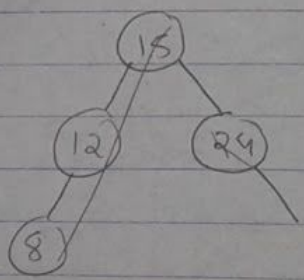
right rotation



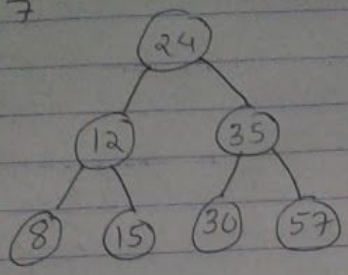
left rotation

left rotation.

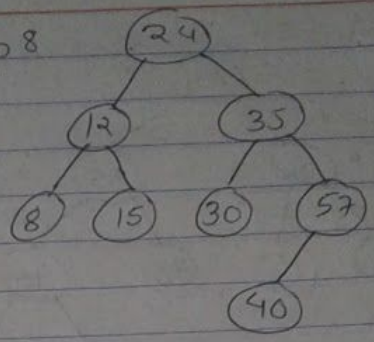




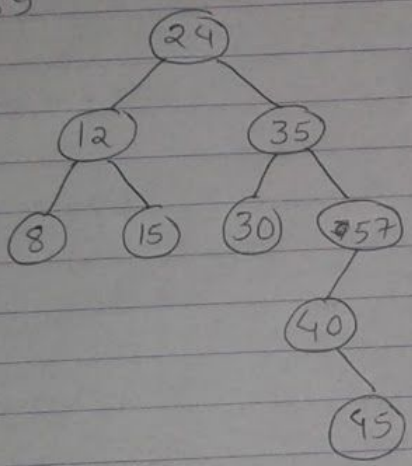
Step 7



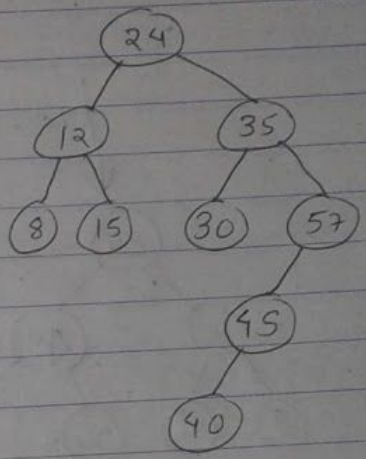
Step 8



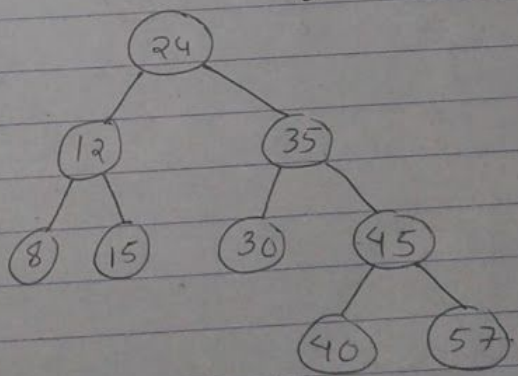
Step 9



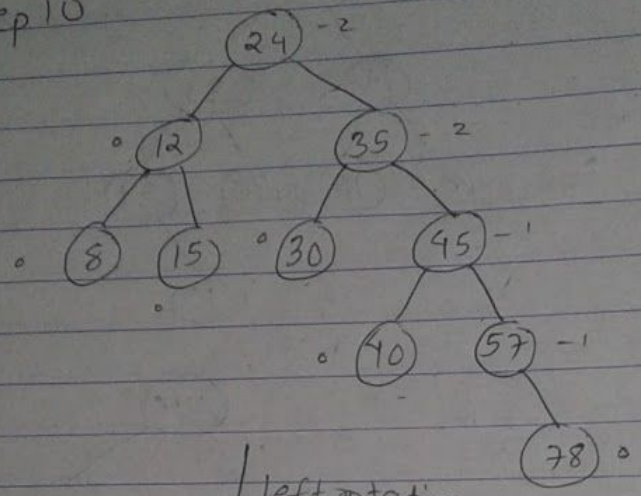
left rotation



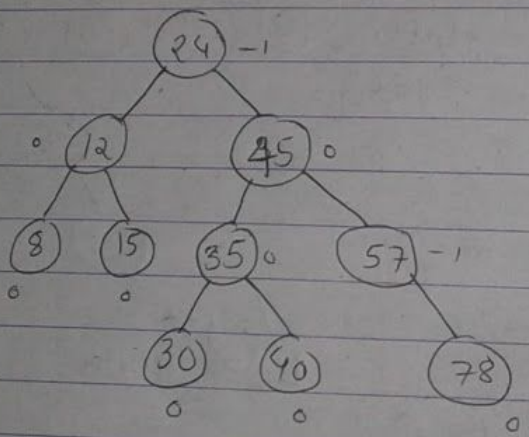
right rotation



Step 10



left rotation



11

Construct a B-tree of order-5 for the following data:
82, 12, 22, 23, 56, 96, 37, 99, 59, 74, 28, 65, 60 & 44

order (m) = 5
maximum keys = m-1 = 4
minimum key for root node = 1
minimum key for non-root node = $\left(\frac{m-1}{2}\right)$ = 2

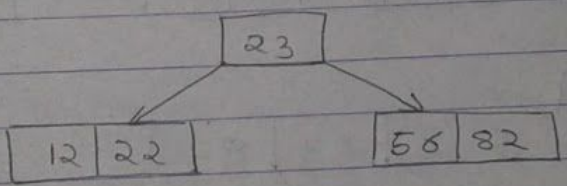
Insert 82 | 82 |

Insert 12 | 12 | 82 |

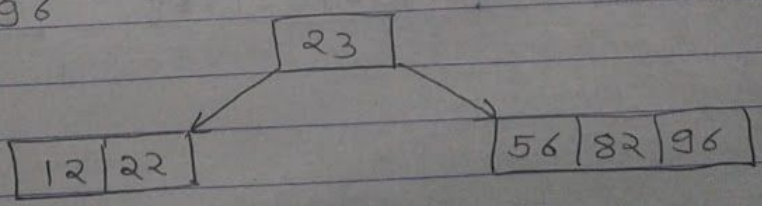
Insert 22 | 12 | 22 | 82 |

Insert 23 | 12 | 22 | 23 | 82 |

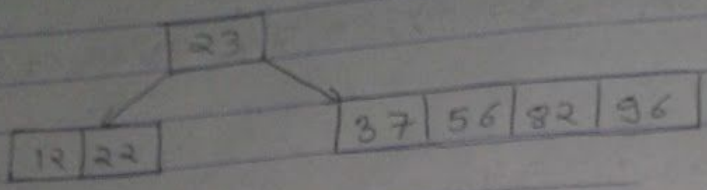
Insert 56 | 12 | 22 | 23 | 56 | 82 | Node overflow



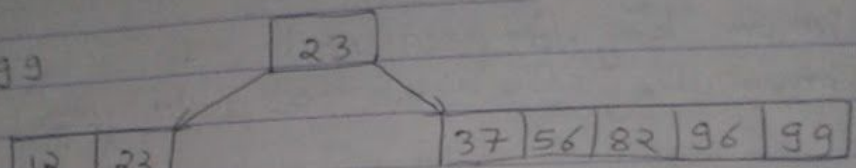
Insert 96



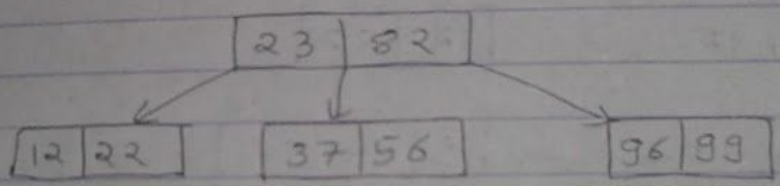
Insert 37



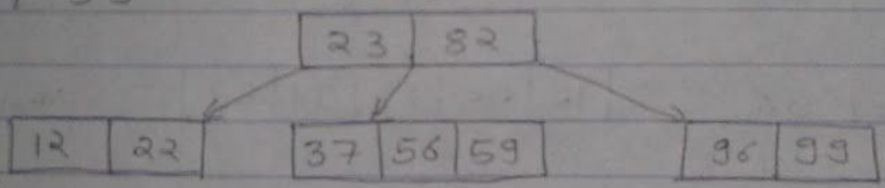
Insert 99



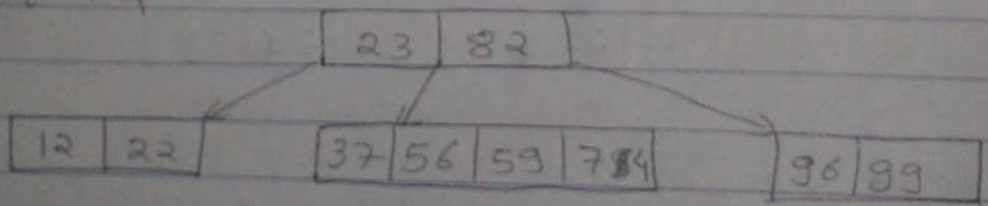
Node overflow



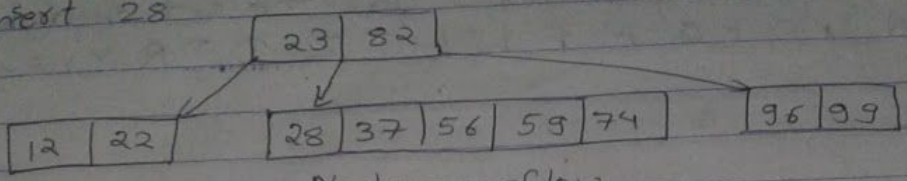
Insert 59



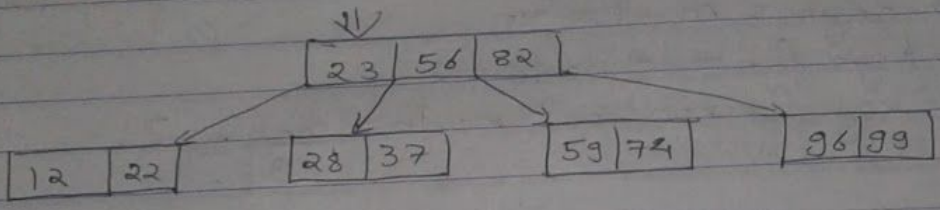
Insert 79



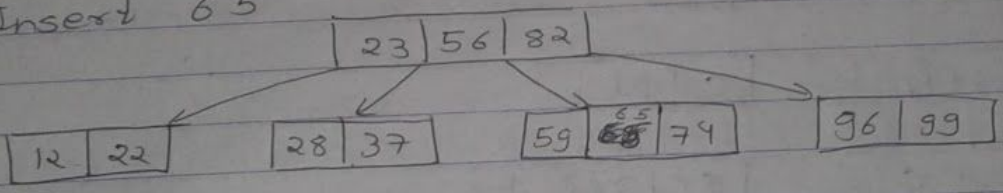
Insert 28



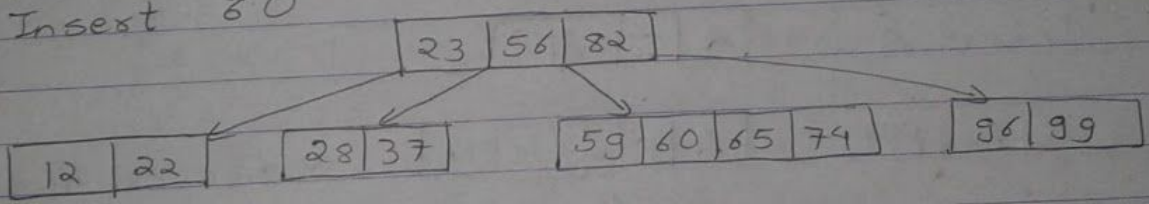
Node overflow



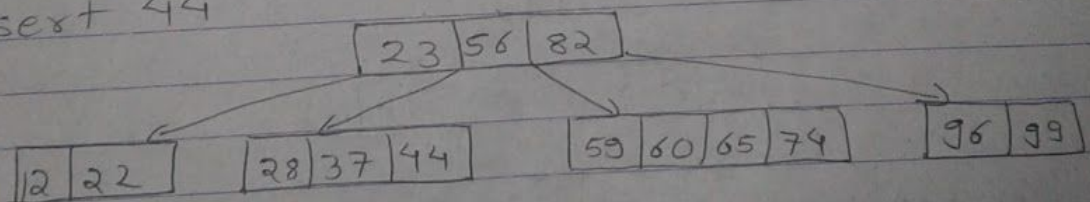
Insert 65



Insert 60



Insert 44



2) Construct B-tree of order 5 for the set of data
C N G A H E K Q M F W L T Z D P R X Y S.

Insert E

$m = 5$

maximum keys $= m - 1 = 4$

minimum keys for root node $= 1$

minimum key for non root node $= \frac{m-1}{2} = 2$.

Insert C [C]

Insert N [C | N]

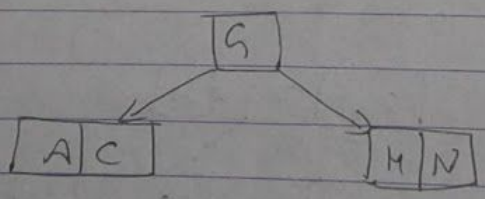
Insert G [C | G | N]

Insert A [A | C | G | N]

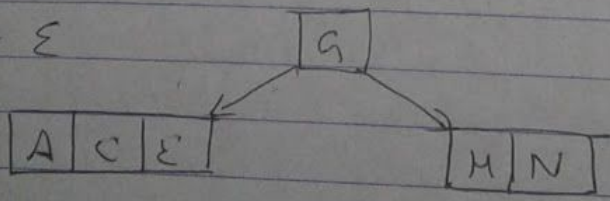
Insert H [A | C | G | H | N] Node overflow

⇓

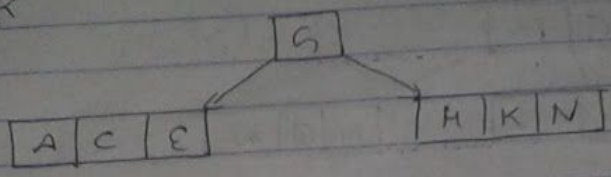
To



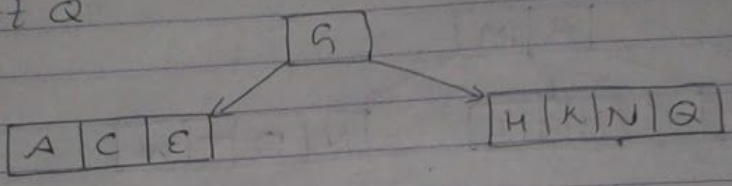
Insert E



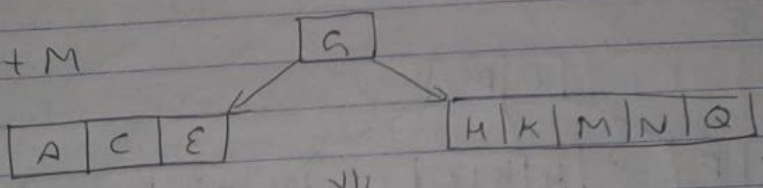
Insert K

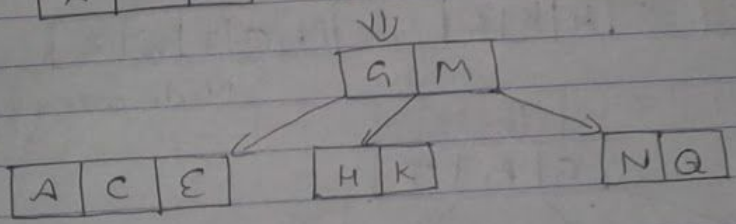


Insert Q

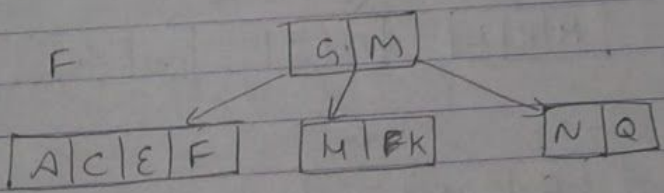


Insert M

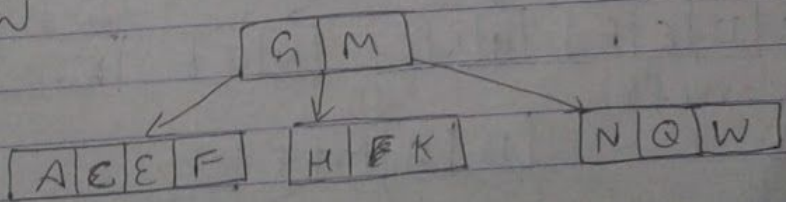




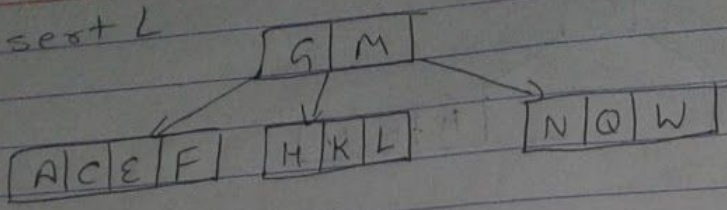
Insert F



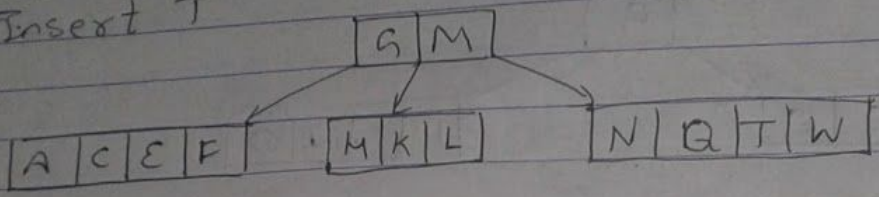
Insert W



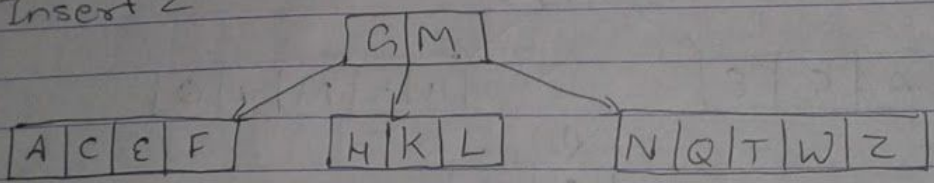
Insert L



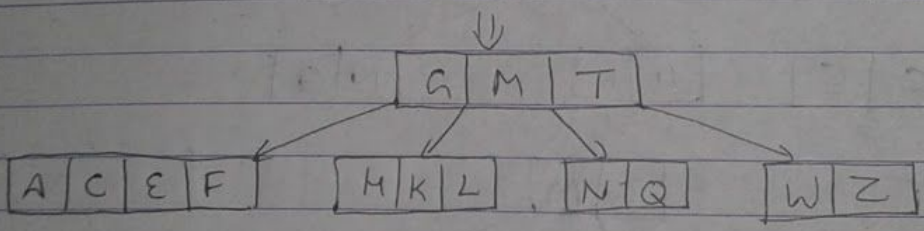
Insert T



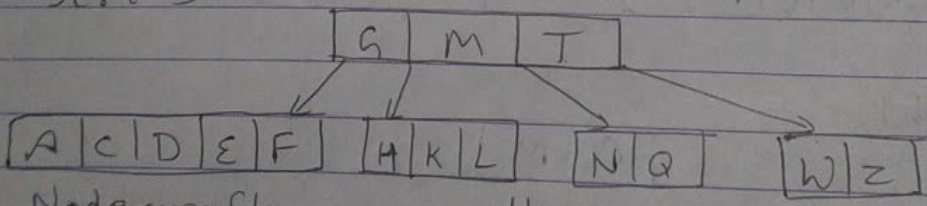
Insert Z



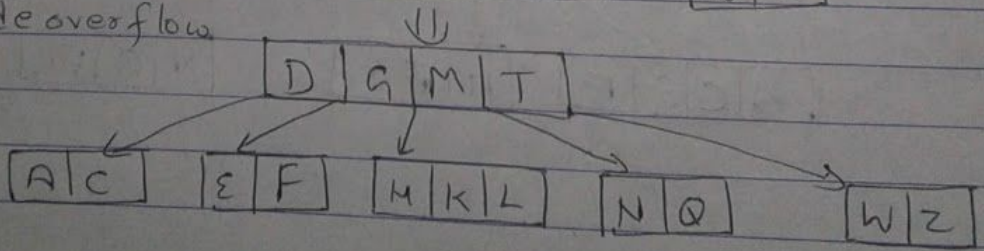
Node overflow



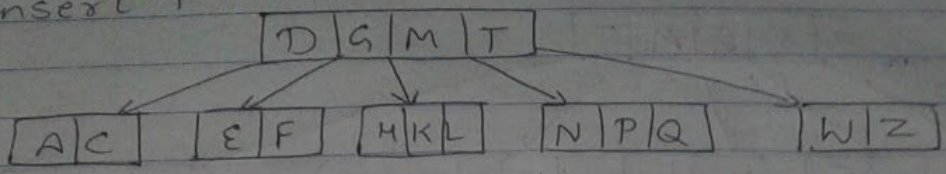
Insert D



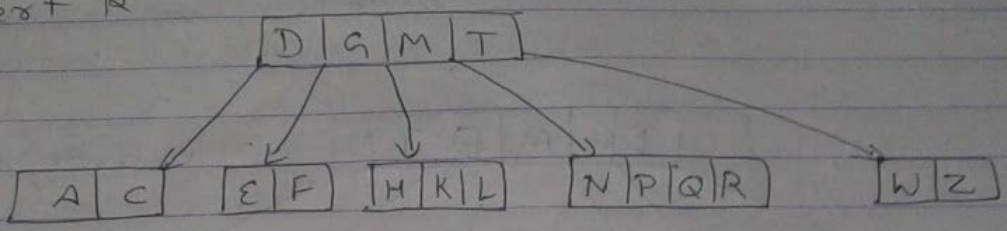
Node overflow



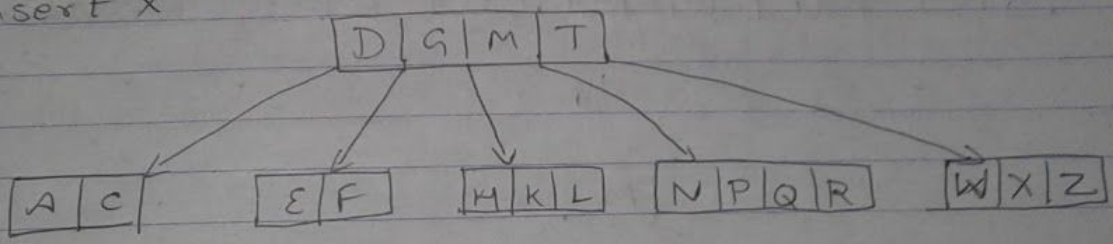
Insert P



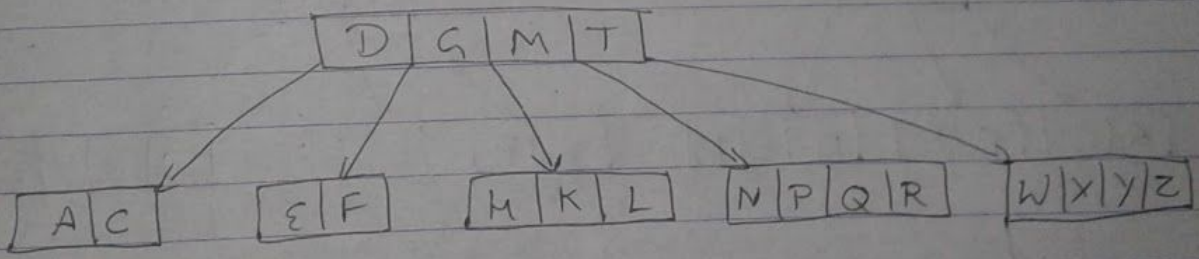
Insert R



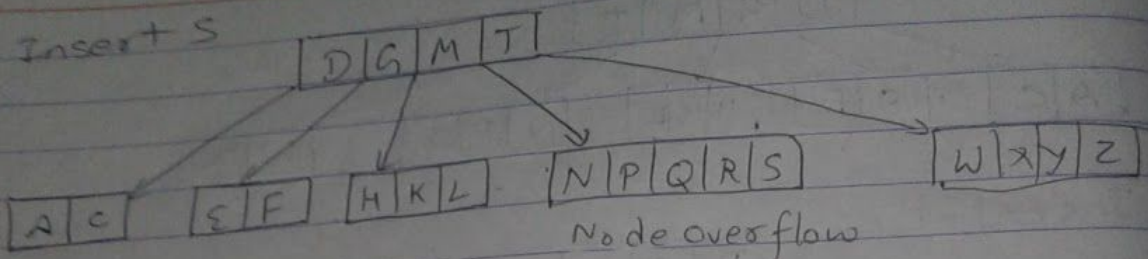
Insert X



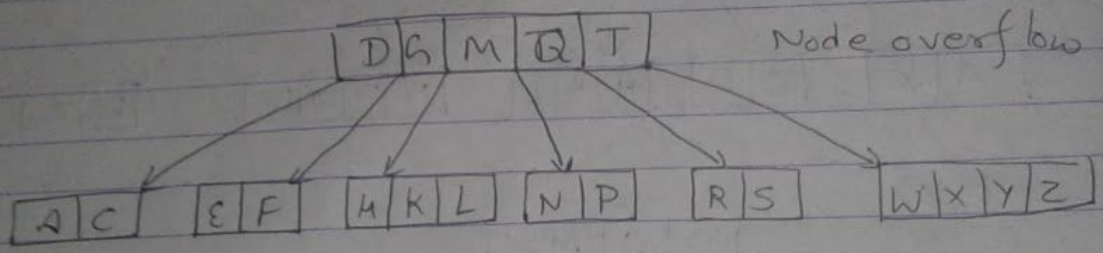
Insert Y



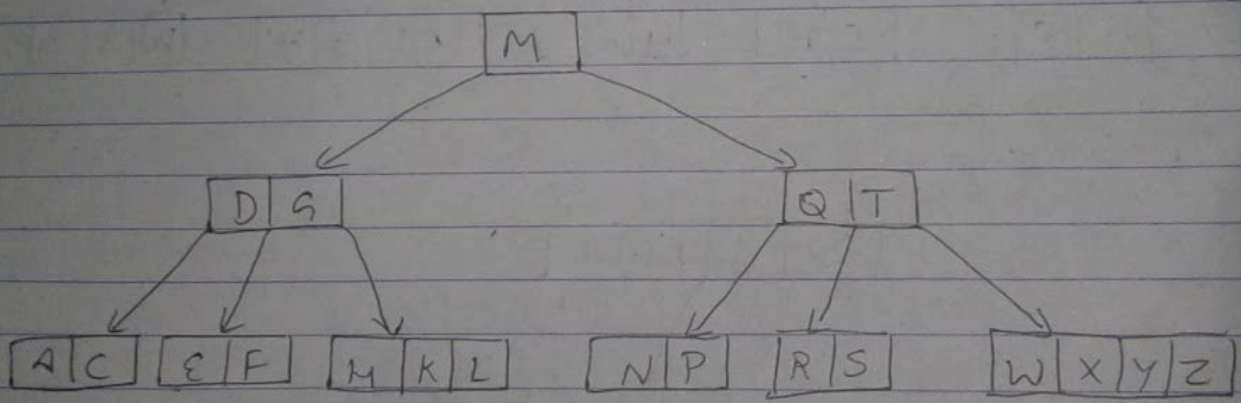
Insert S



Node overflow



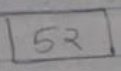
Node overflow



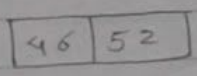
③ Construct a B-tree of order 3 for:
52, 46, 27, 81, 90, 108, 72, 110, 35, 115, 121 & 86

$m = 3$
maximum keys $= m - 1 = 2$
minimum key for root node $= 1$
minimum key for non-root node $= \frac{m-1}{2} = 1$

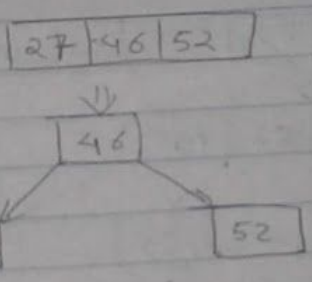
Insert 52



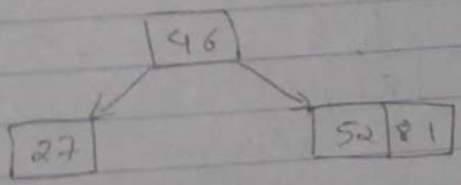
Insert 46



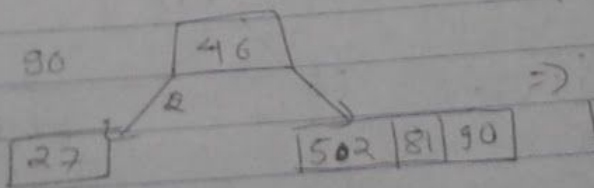
Insert 27.



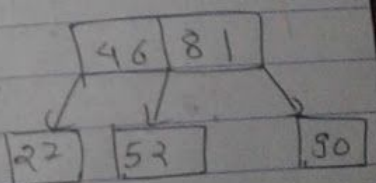
Insert 81



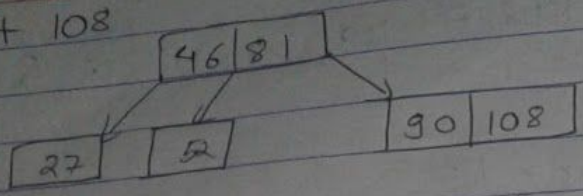
Insert 80



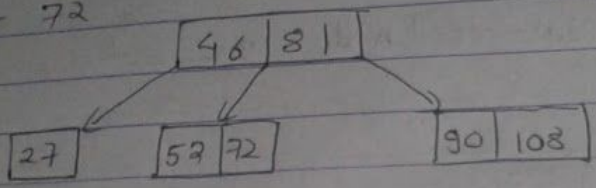
=>



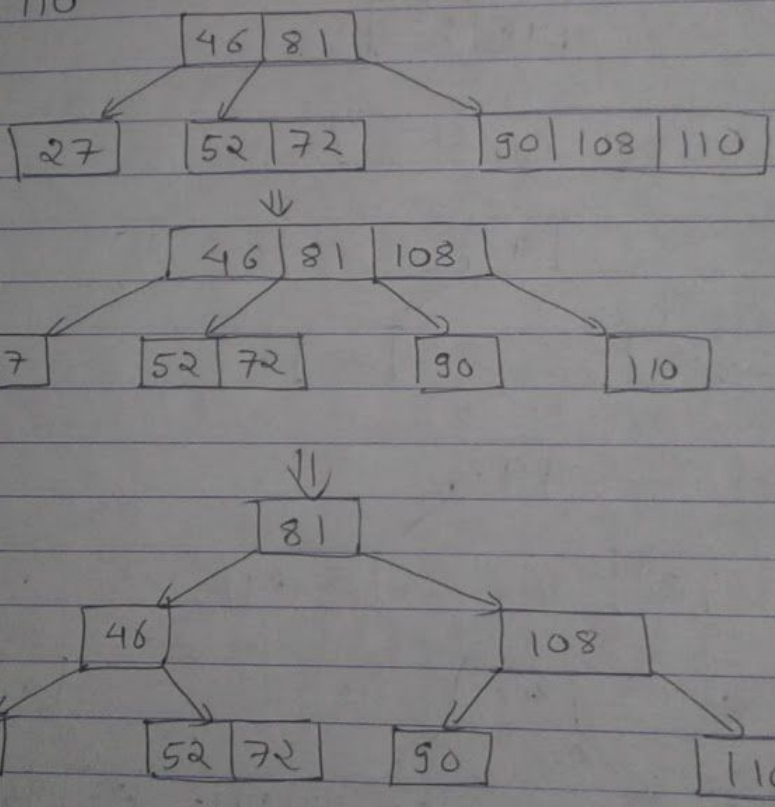
Insert 108



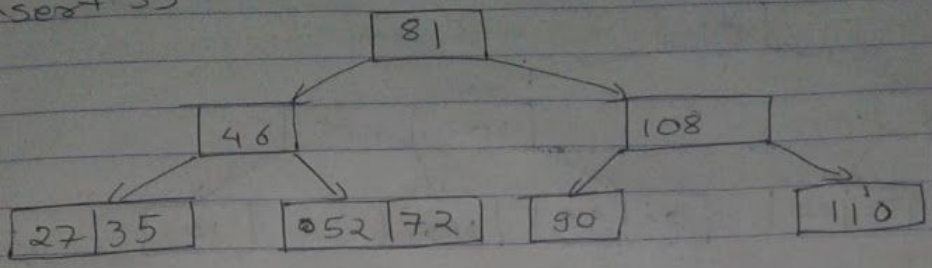
Insert 72



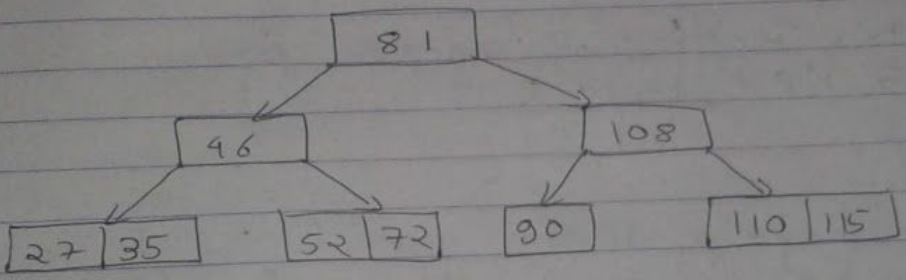
Insert 110



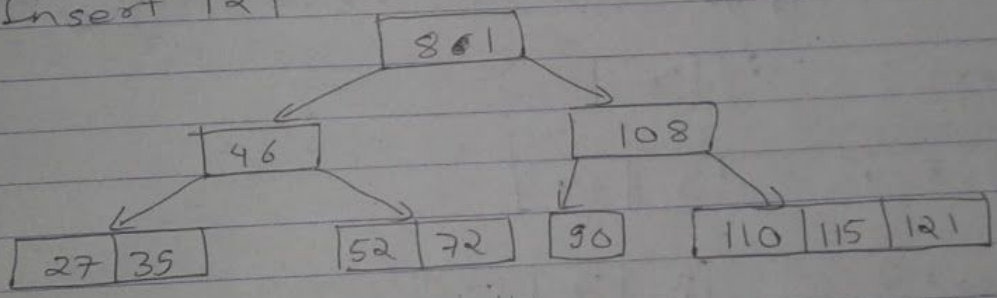
Insert 35



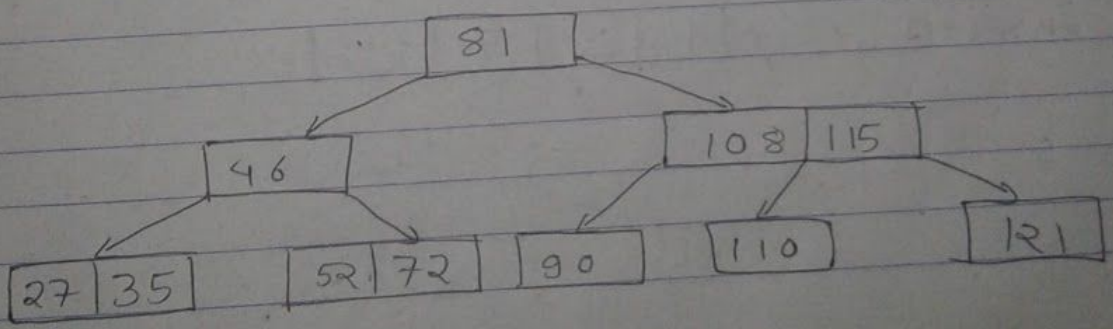
Insert 115



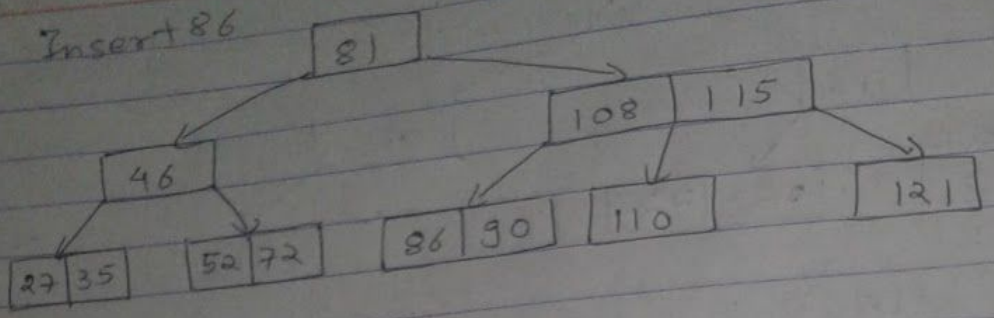
Insert 121



Node overflow



Insert 86



//

④ Construct a B-tree of order 5 for
3, 14, 7, 1, 8, 5, 11, 17 & 13

∴ $m = 5$
maximum key $= m - 1 = 4$
minimum key for root node $= 1$
minimum key for non-root node $= \frac{m-1}{2} = 2$

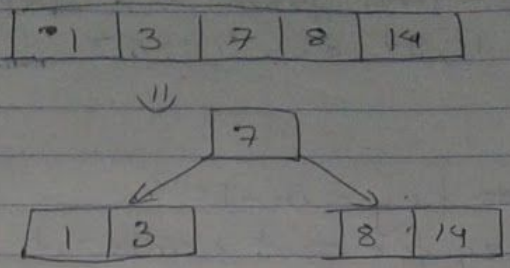
Insert 3 | 3 |

Insert 14 | 3 | 14 |

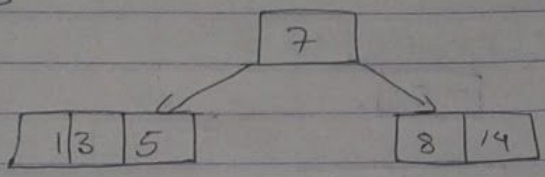
Insert 7 | 3 | 7 | 14 |

Insert 1 | 1 | 3 | 7 | 14 |

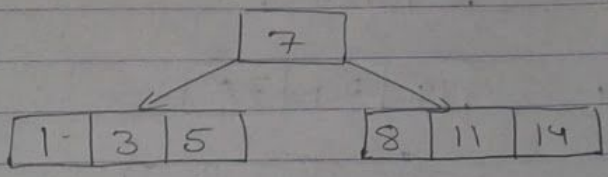
Insert 8



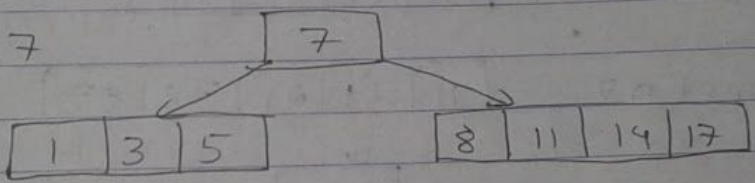
Insert 5



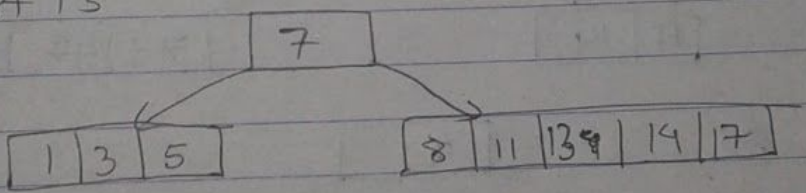
Insert 11



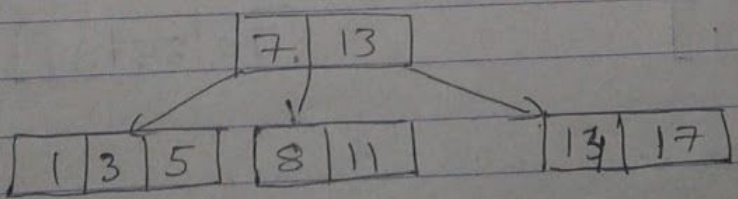
Insert 17



Insert 13



=>



⑤ Construct B-tree from the dataset given,
78, 21, 14, 11, 97, 85, 74, 63

Let $m = 5$
maximum key $= m - 1 = 4$
minimum key for root node $= 1$.
minimum key for non-root node $= \frac{m-1}{2} = 2$

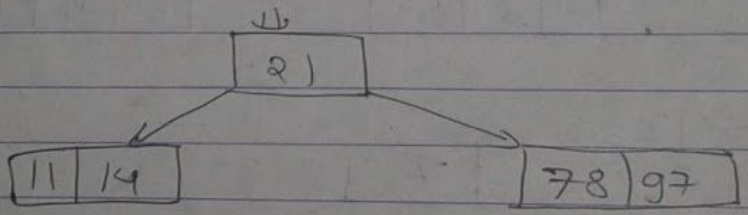
Insert 78 | 78 |

Insert 21 | 21 | 78 |

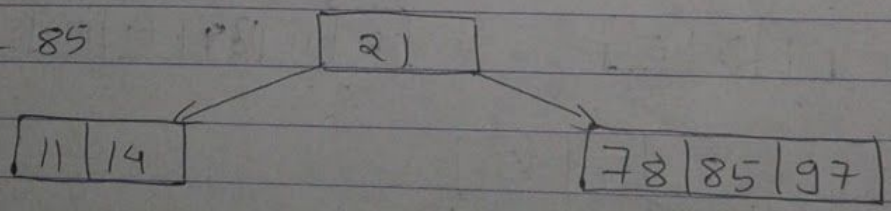
Insert 14 | 14 | 21 | 78 |

Insert 11 | 11 | 14 | 21 | 78 |

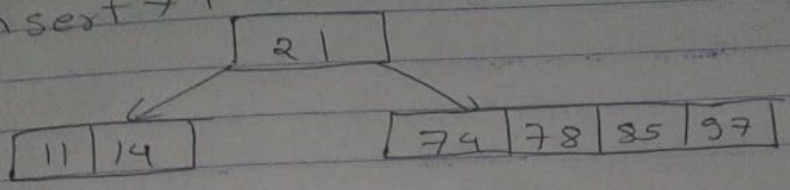
Insert 97 | 11 | 14 | 21 | 78 | 97 |



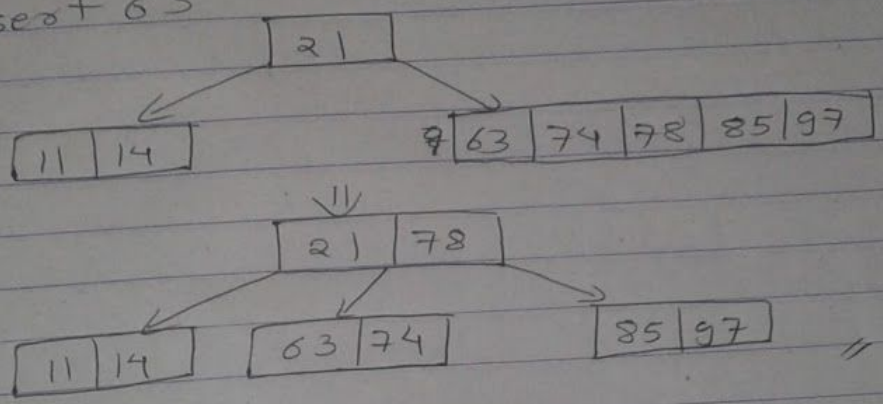
Insert 85



Insert 74



Insert 63



PAGE NO.:
DATE:

Construct a B-tree of order 3

1, 2, 3, 4, 5, 6, 7

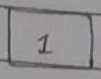
$m = 3$

maximum key $= m - 1 = 2$

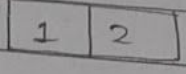
min no. of nodes in root node $= 1$

min no. Key for non root node $= \frac{m-1}{2} = \frac{3-1}{2} = 1$

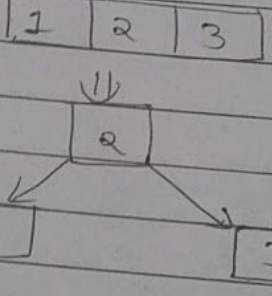
Insert 1



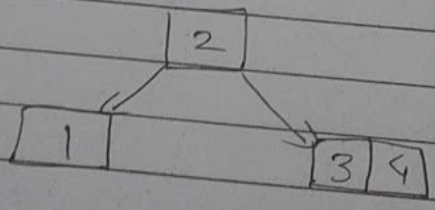
Insert 2



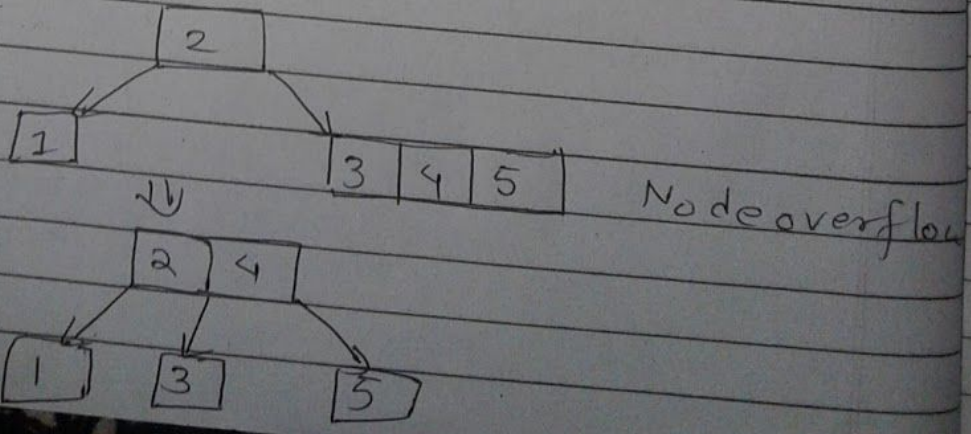
Insert 3



Insert 4

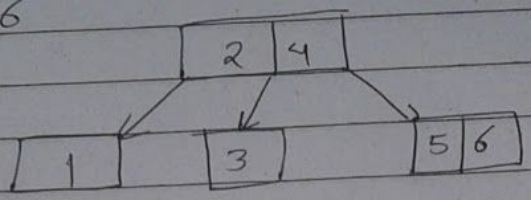


Insert 5

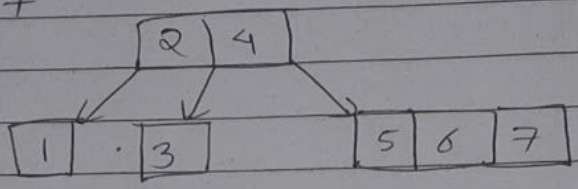


PAGE NO. :
DATE :

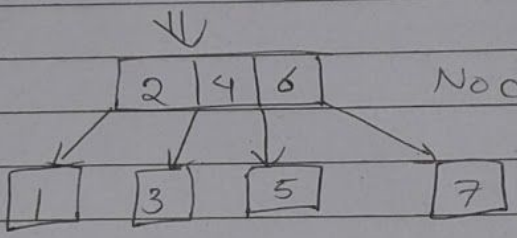
Insert 6



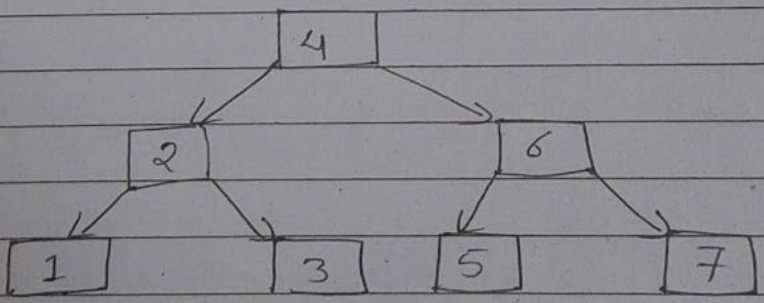
Insert 7



Node overflow



Node overflow

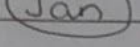


PAGE NO. :
DATE :

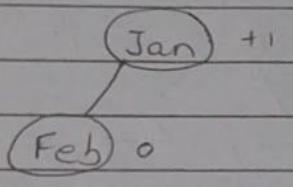
Create AVL tree in Alphabetical order

Jan, Feb, March, April, May, June, July, Aug, Sep, Oct, Nov, Dec.

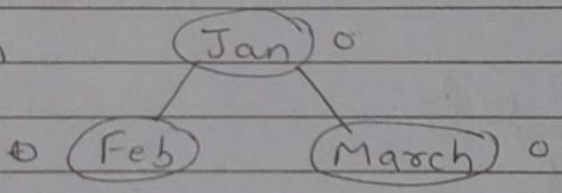
(1) Insert Jan



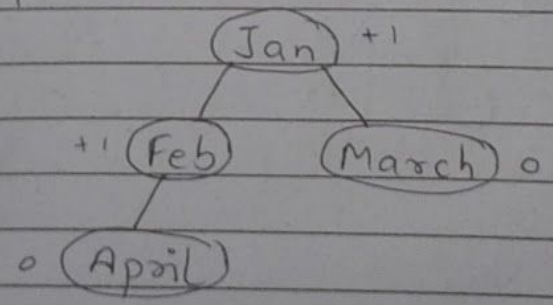
(2) Insert Feb



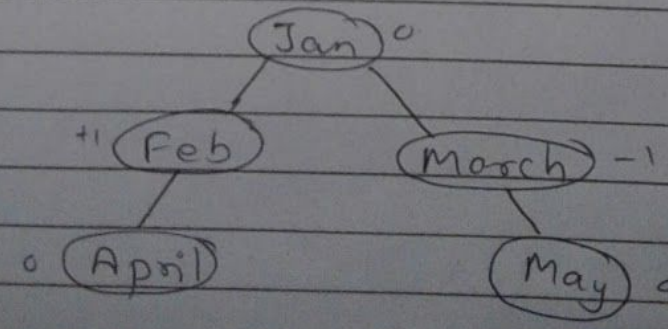
(3) Insert March



(4) Insert April

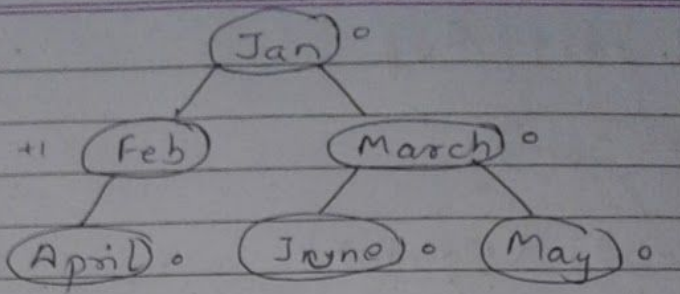


(5) Insert May

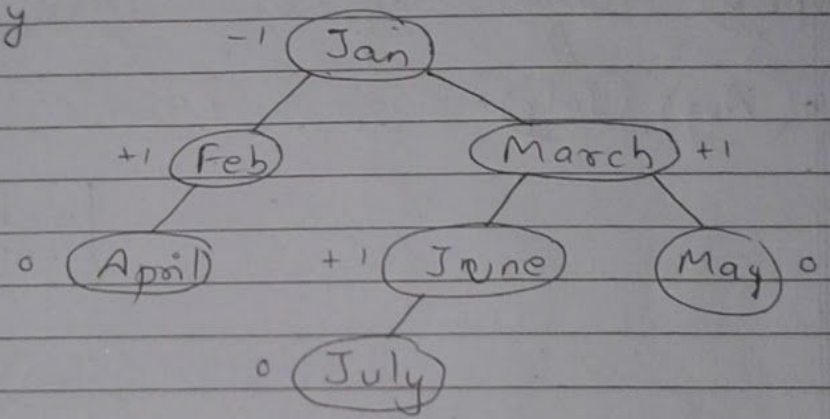


PAGE NO. :
DATE :

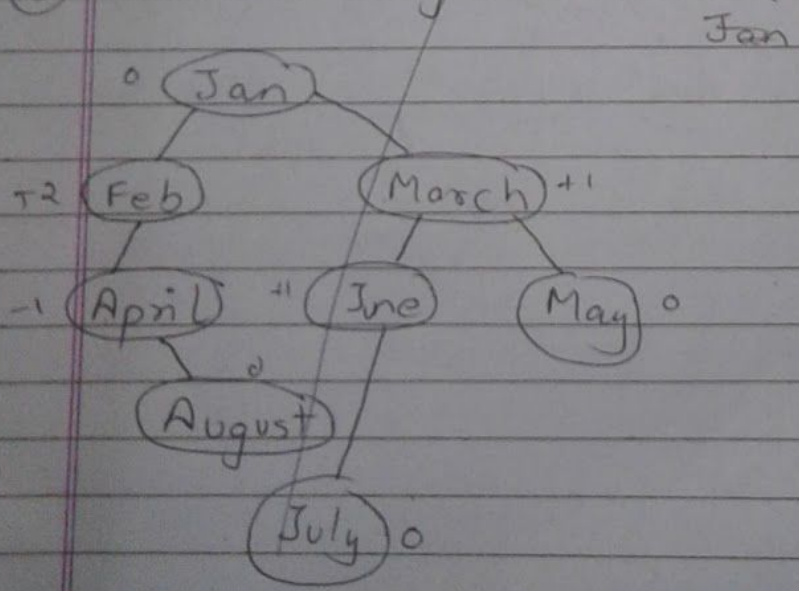
(6) Insert June



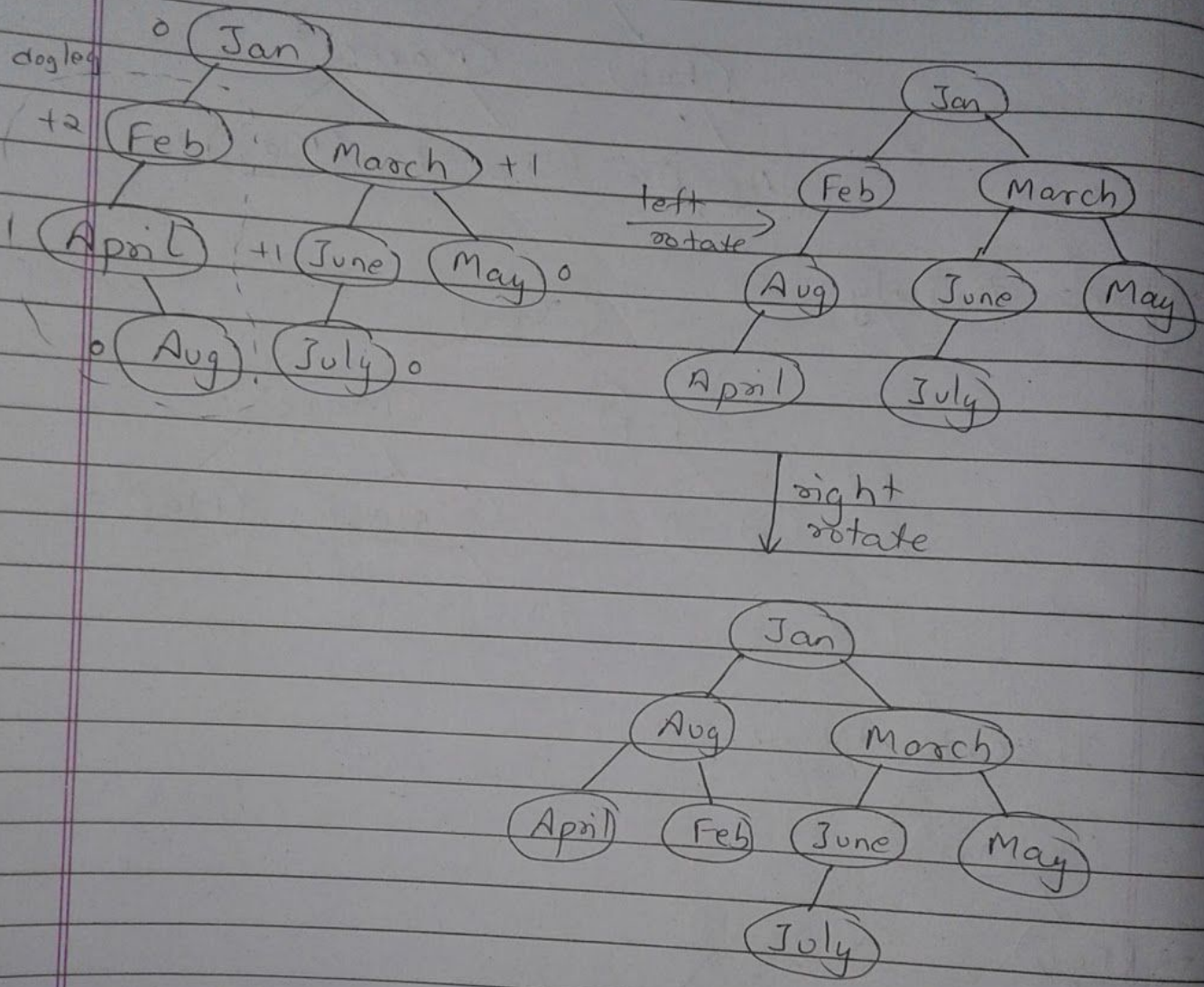
(7) Insert July



(8) Insert August.

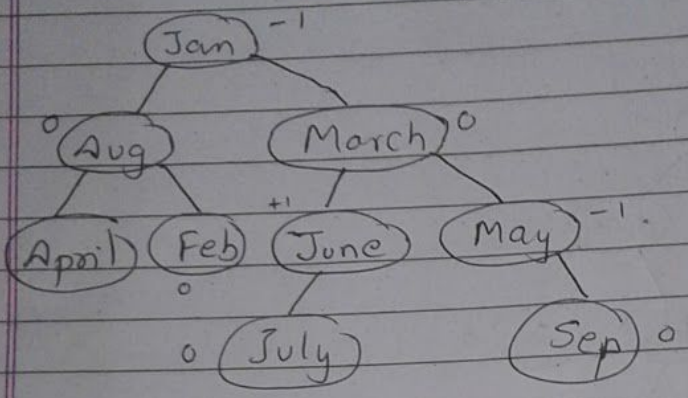


(8) Insert Aug

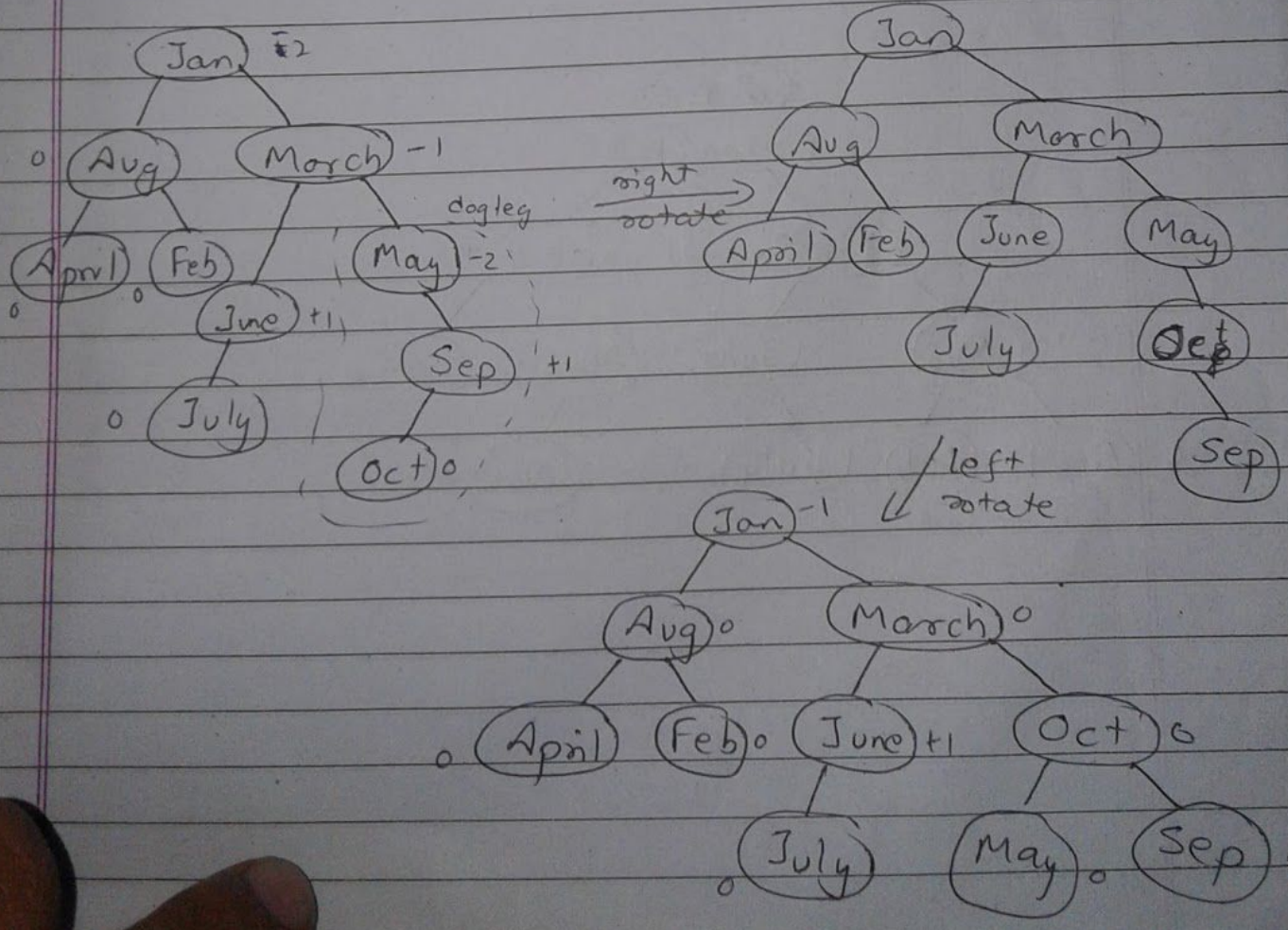

dogleg
0
Jan
+2
Feb
March
+1
-1
April
+1
June
May
0
0
Aug
July
0
left rotate
Jan
Feb
March
Aug
June
May
April
July
right rotate
Jan
Aug
March
April
Feb
June
May
July

PAGE NO. :
DATE :

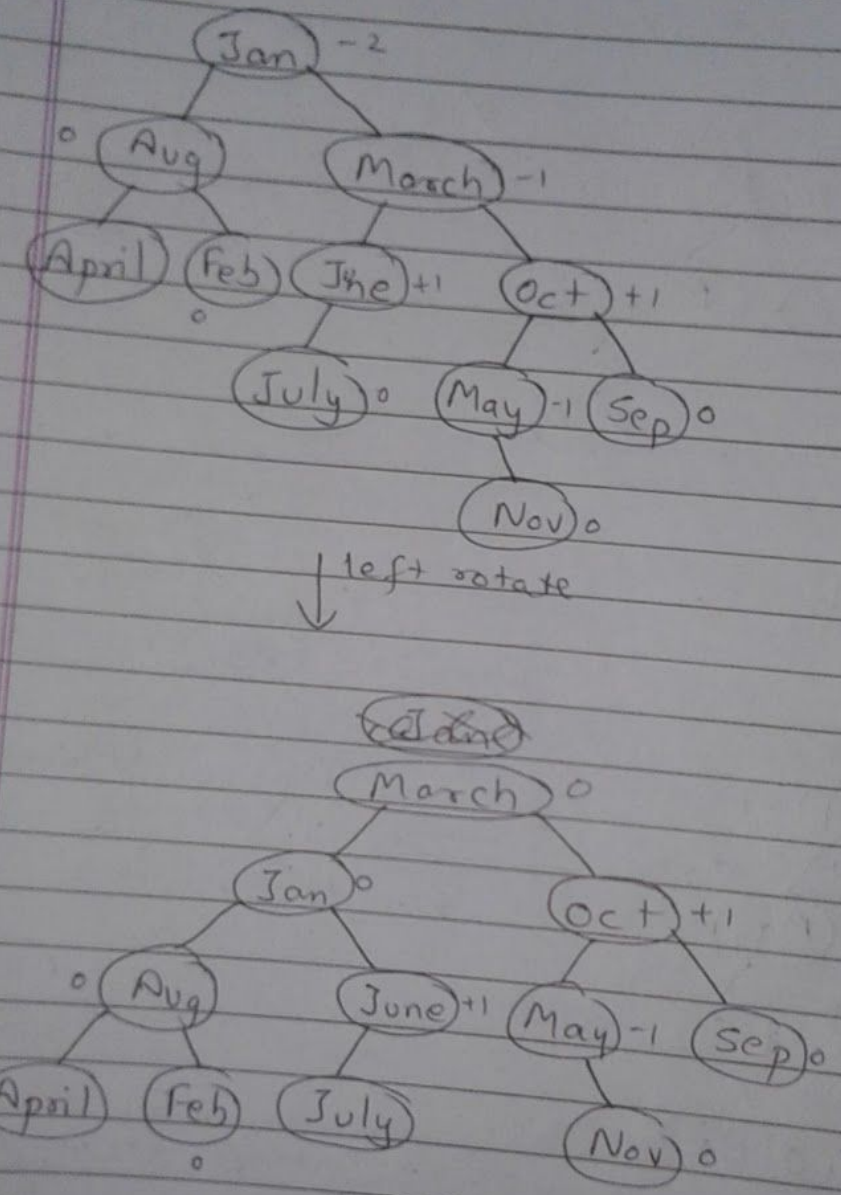
(9) Insert Sep



(10) Insert Oct

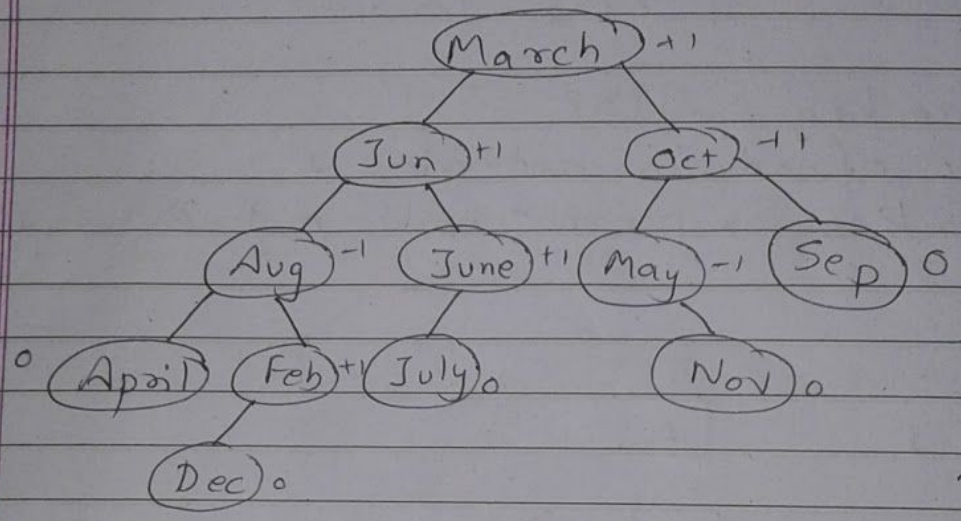


(11) Insert Nov



PAGE NO. :
DATE :

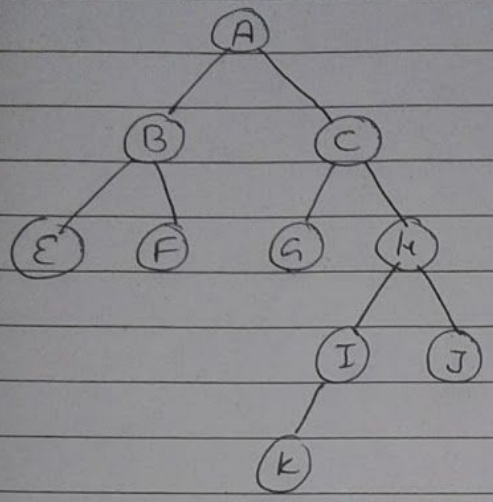
(12) Insert Dec



PAGE NO. :
DATE :

Tree.

(1)

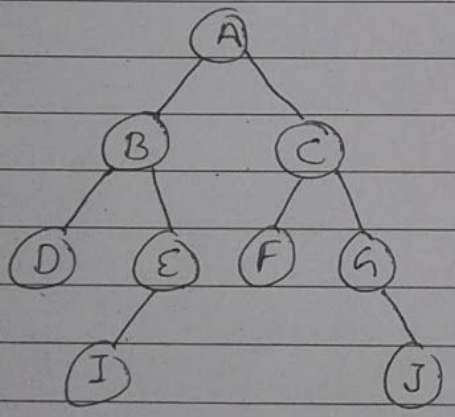


Preorder: A B E F C G H I K J

Inorder: E B F A G C K I H J

Postorder: E F B G K I J H C A

(2)



Preorder: A B D E J F C F G J

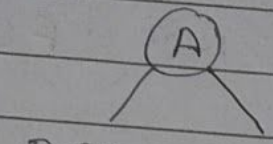
Inorder: D B I E A F C G J

Postorder: D I E B A F G J G C A

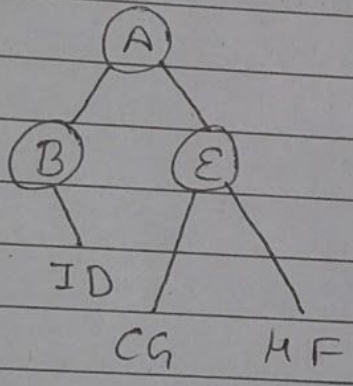
PAGE NO. :
DATE :

Postorder I D B G C H F E A
Inorder B I D A C G E H F

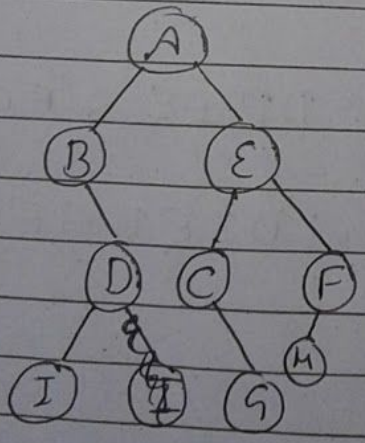
(1)



(2)

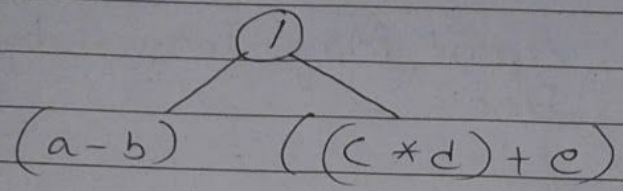


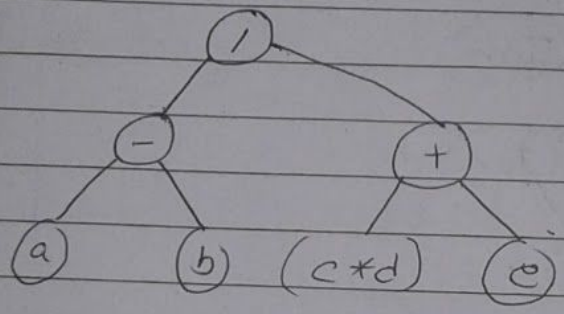
(3)



PAGE NO. :
DATE :

(a-b) / ((c * d) + e)

① 

② 

③