__/__/____

27.02.2024

B+ TREES → Search tree

BST
↳ Ordering property
Height: $\log_2 n$
No. of nodes (n) is very high ⇒ Search time will be very high

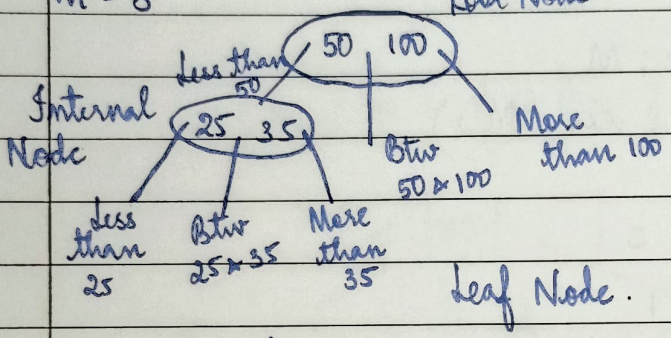
Multiway (M-way) Trees } To reduce high of a tree
↳ Order of tree }

Search Tree

For a BST ⇒ M = 2

key = M − 1 = 2 − 1 = 1

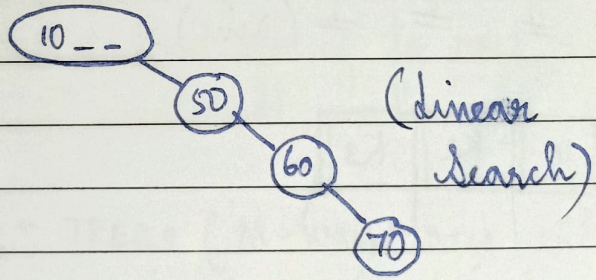
M = 3



Search time becomes better

M can start with 3 and go on to any value.

M = 4

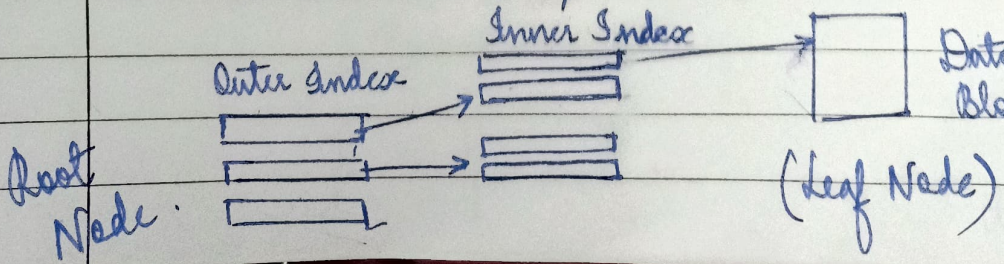


(linear search)

B & B+ tree → Balanced M-way tree
↳ MORE ADVANTAGE

Harddisk → Chunk of blocks

MULTILEVEL INDEXING → Faster access.



B+ tree → Search tree
M-way tree
all leaf node at same level.
Pointer for all block available at leaf node.

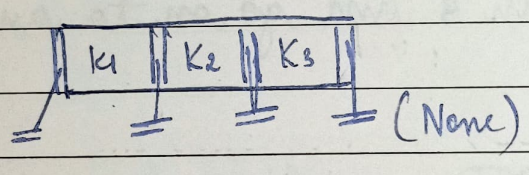
NOT APPLICABLE TO A LEAF NODE

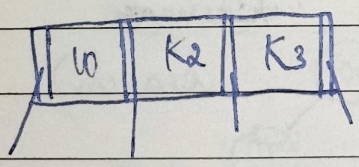
M-way :

Keys → Max : M-1 keys.
Min : ceil(M/2) - 1

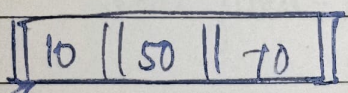
Children → Max : M
Min : ceil(M/2)

M=4 Key = m-1 = 3
Min = 1.



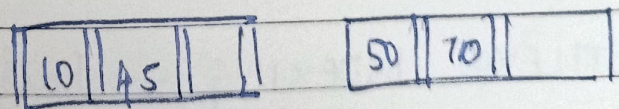


Keys → descending order.



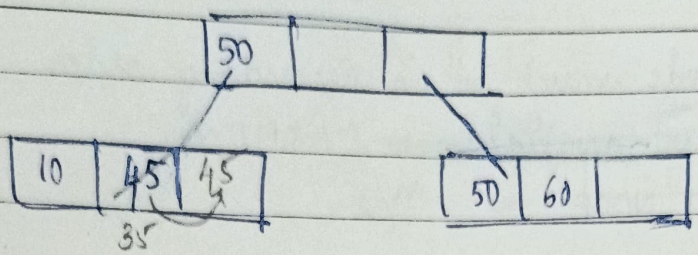
(One level of index)

10 45 50 70 (Not possible (overflow))



Overflow
↓
Leaf Node
↓
Split

B tree → Keys need not be in leaf
has to be only there ~~in Internal~~ nodes

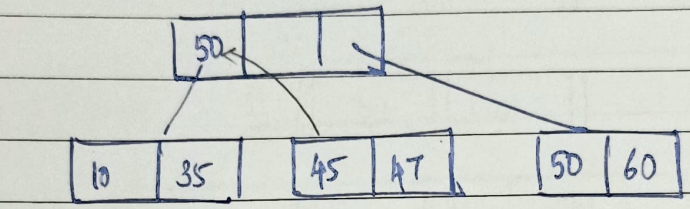


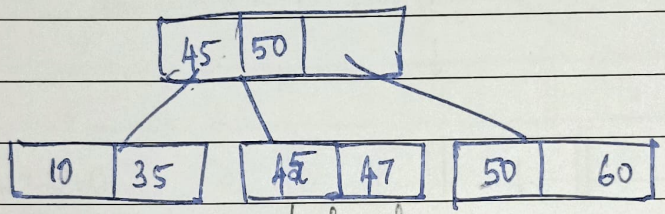
Leaf node has exact data,

Insert 35

Insert 47 Overflow

All these data is organised linearly in hard disk.
Leaf node → Singly linked list





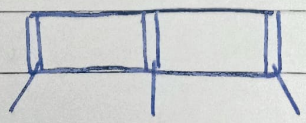
Leaf node

Internal node ⇒ Node going to root must not be in internal node.

29.02.2024 PROPS OF B+ TREES (M-way trees with foll. props)

1) Order m
2) Children max: m min: ceil(m/2)
3) Keys max: m-1 min: ceil(m/2)-1
4) All keys should be there in leaf node.
5) All leaf same level.

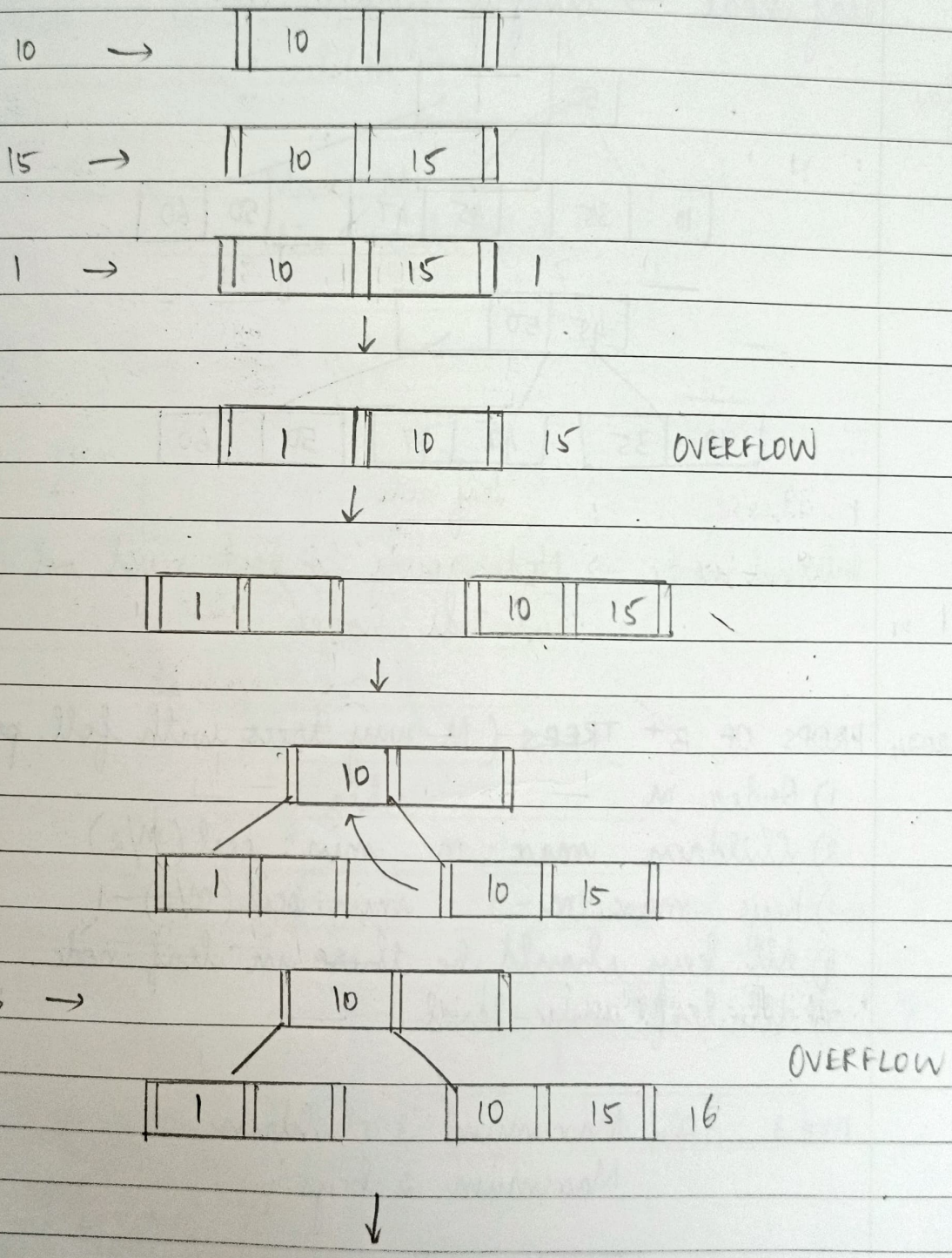
m=3 → Maximum 3 children
Maximum 2 keys.



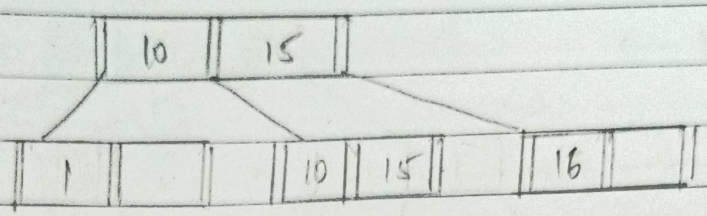
__/__/____

Keys on the node must be in ascending order.
More keys than available ⇒ OVERFLOW.
⇒ SPLIT NODE ⇒ $m/2$.

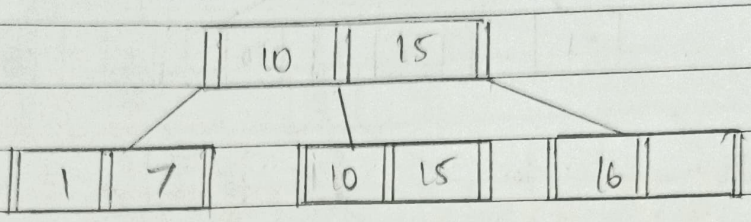
10, 15, 1, 16, 7, 25, 23, 17



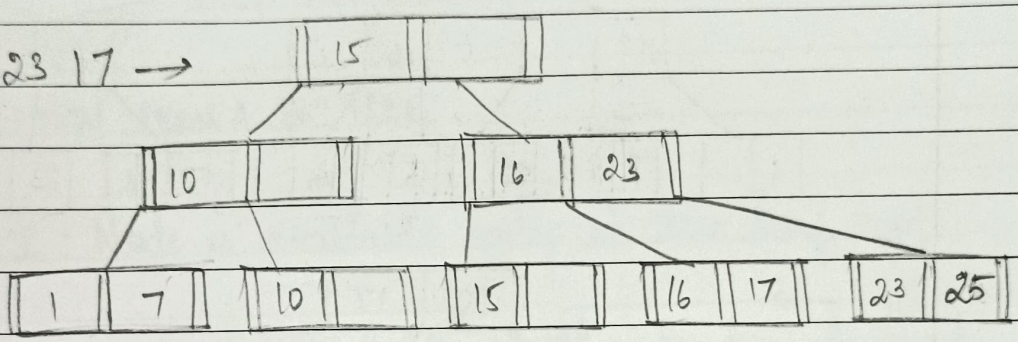
__/__/____



7 →



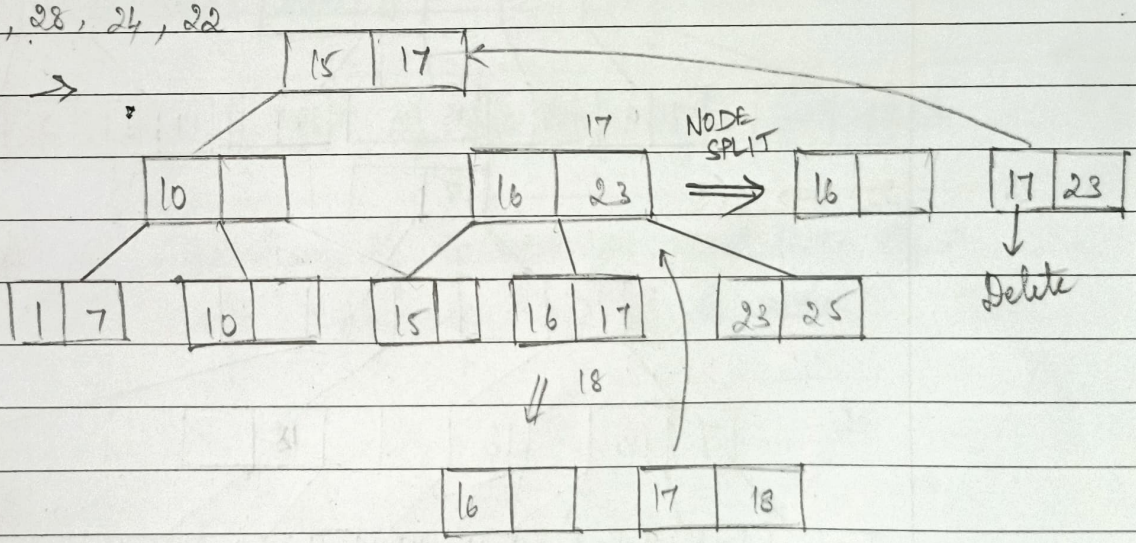
25 23 17 →



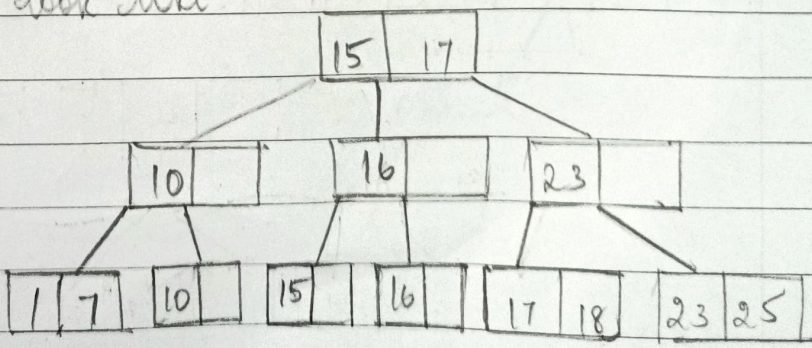
18, 9, 28, 24, 22

18 →

Split on Internal node ⇒ PUSH TO ROOT



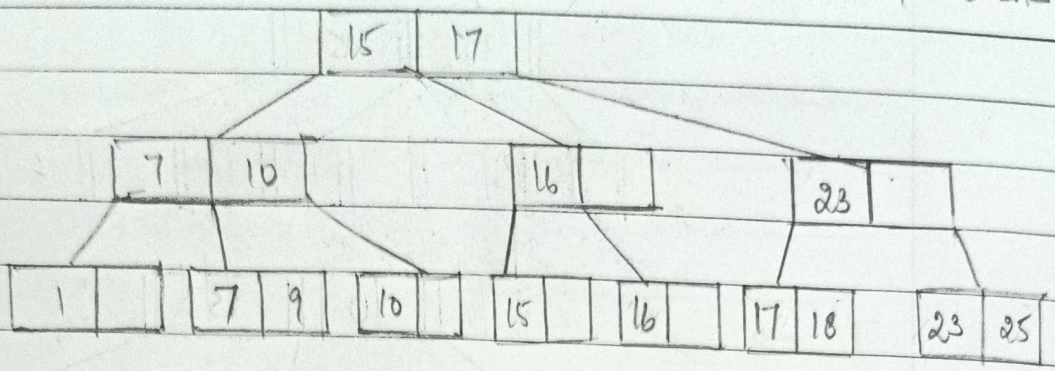
Now this will look like



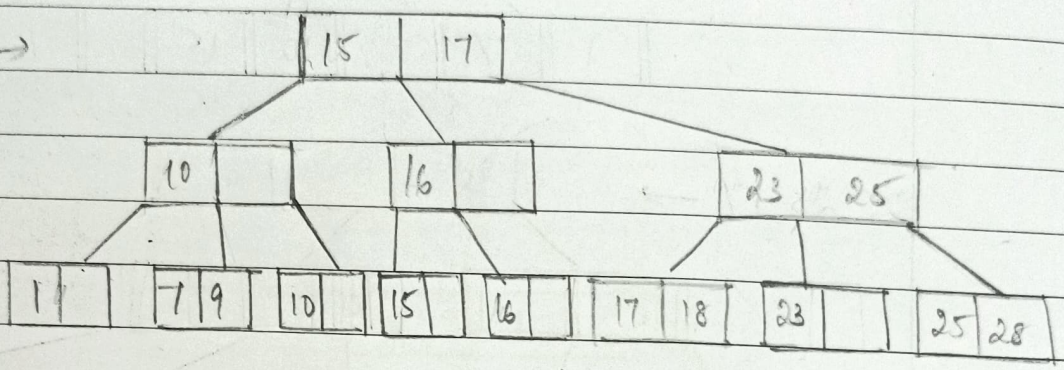
When data increases ⇒

MULTI LEVEL INDEXING INCREASES

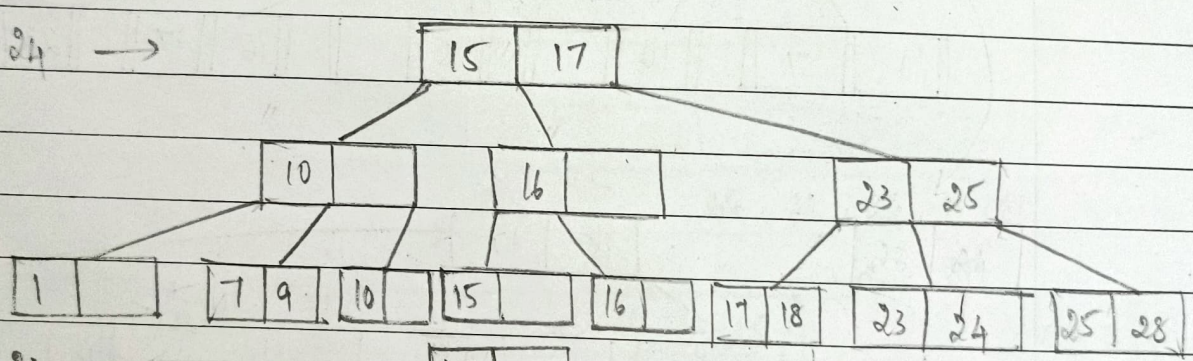
9 →



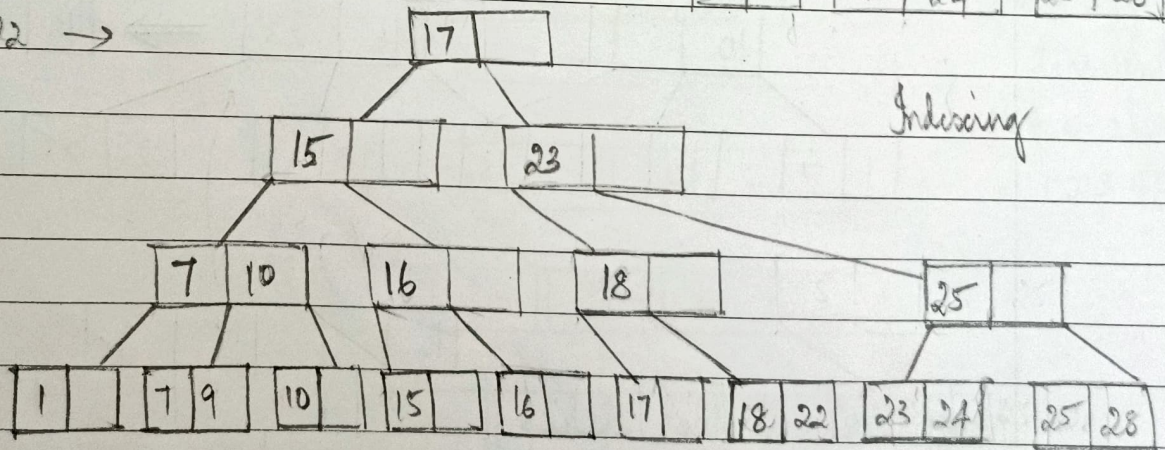
28 →



24 →



22 →



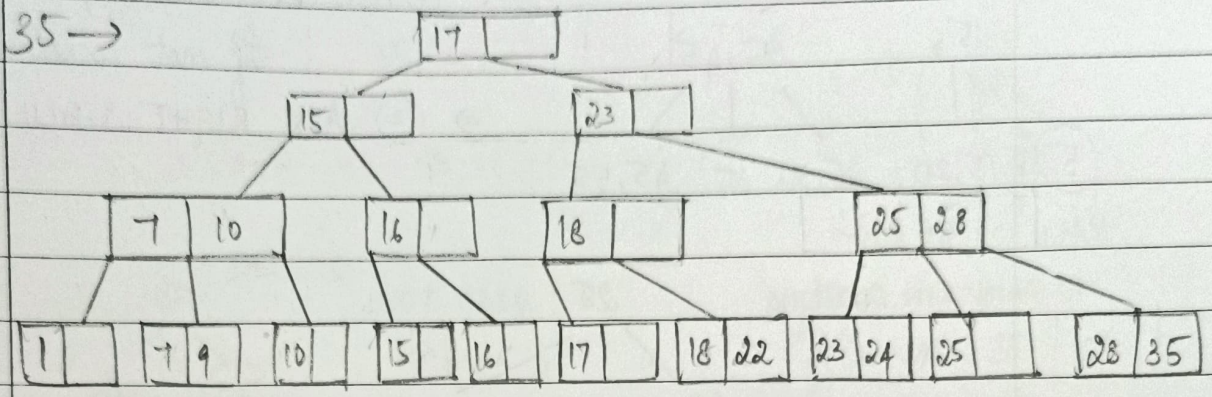
Indexing

AVL → Balanced BST

B^+ TREES → Self Balanced
No rotations

__/__/____

B^+ TREES → Bottom up approach,



01.03.2024. DELETION FROM A B^+ TREE

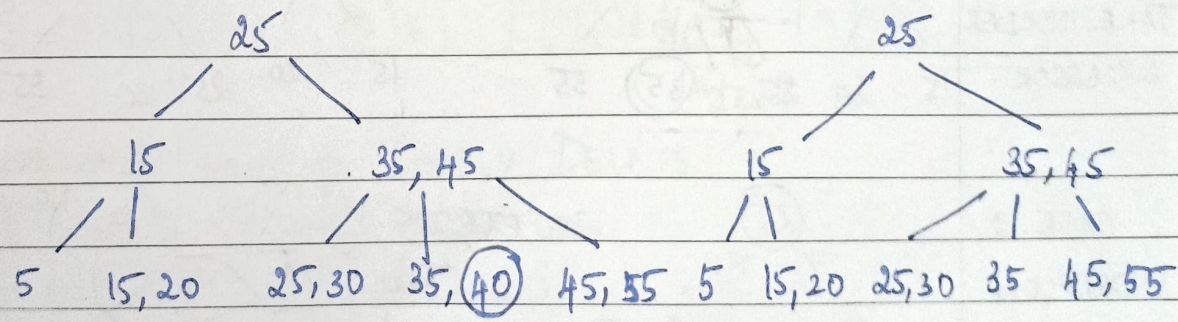
Case I : Node is available only in the leaf node.

Case II : Node is available in the leaf node and internal node

Case III : If the height of the tree gets shrinked

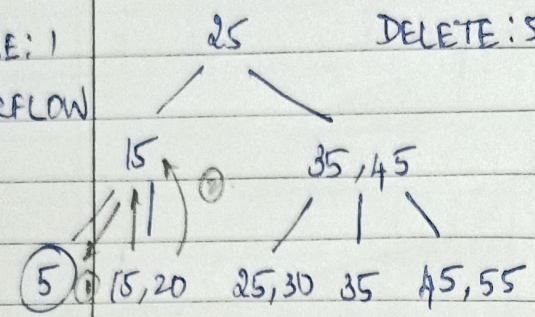
Underflow : Violation of minimum condition of a node (Minimum keys of a node)

CASE: 1
NO UNDERFLOW



__/__/____

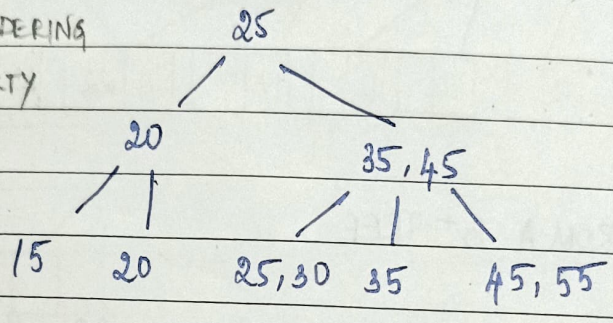
CASE: 1 UNDERFLOW



DELETE: 5 BORROW FROM SIBLING.

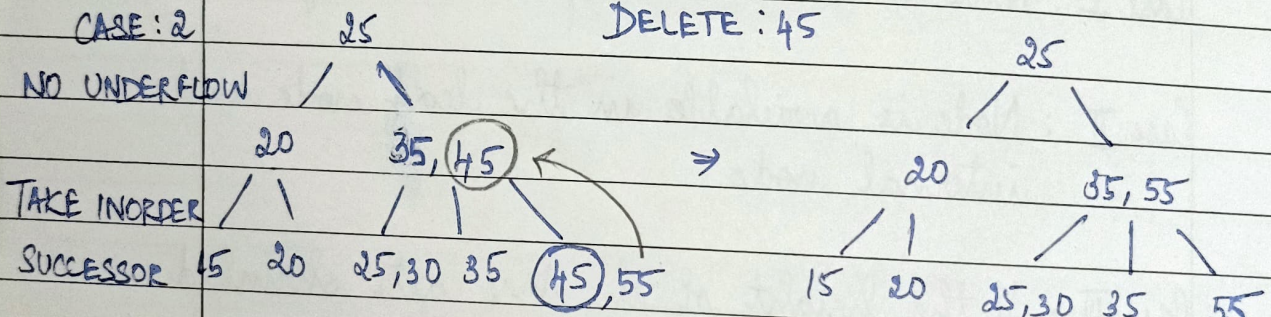
① ASK LEFT SIBLING
If not existing
⇒ ② ASK RIGHT SIBLING.

MAKE 20 AS ROOT TO MAINTAIN ORDERING PROPERTY



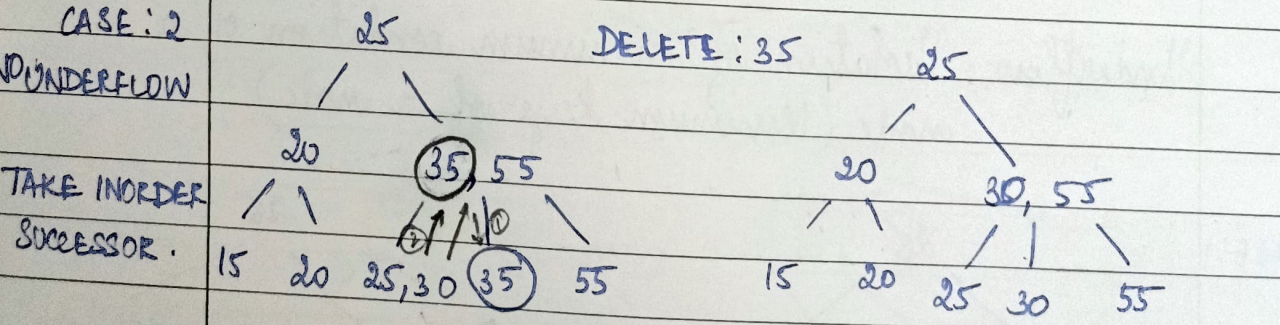
CASE: 2 NO UNDERFLOW

TAKE INORDER SUCCESSOR

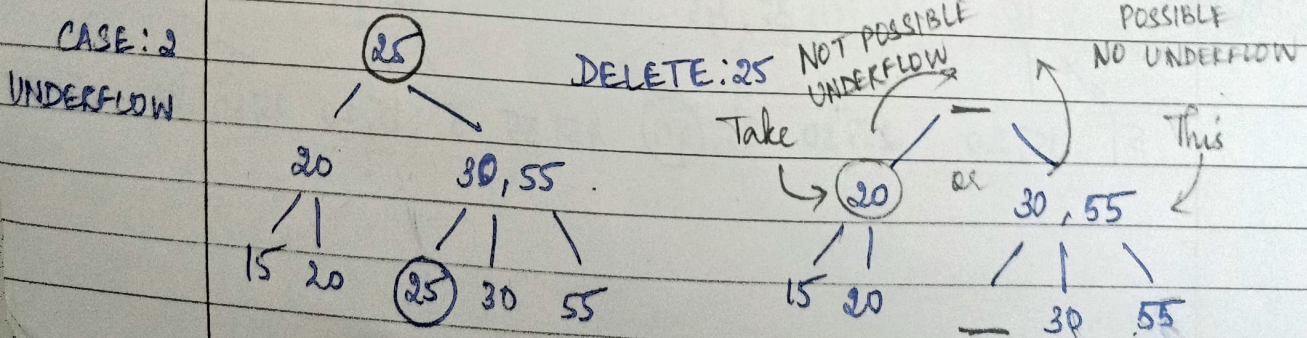


CASE: 2 NO UNDERFLOW

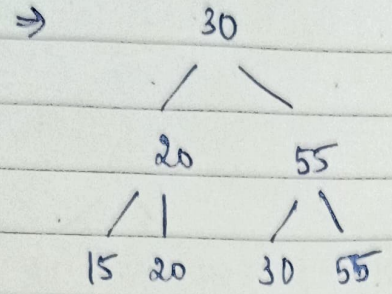
TAKE INORDER SUCCESSOR.



CASE: 2 UNDERFLOW

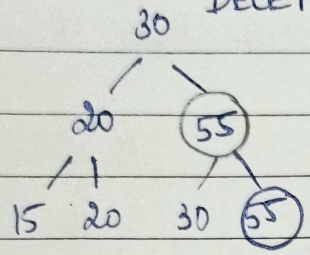


__/__/____



CASE : 3

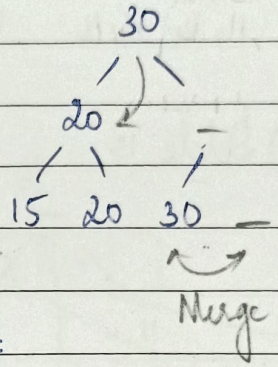
DELETE : 55

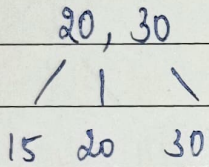


NOT CASE : 1

CASE : 2

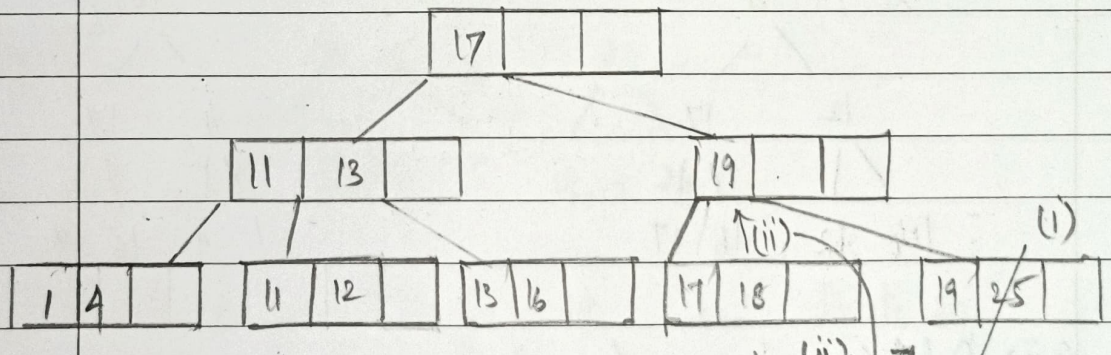
UNDERFLOW





HEIGHT OF TREE REDUCES BY 1.

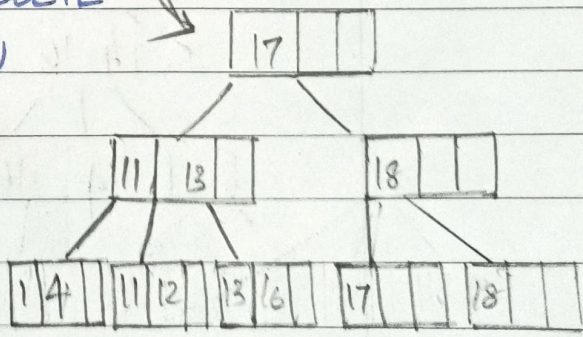
Que: Order : 4



(i) Delete 25 ⇒ SIMPLY DELETE

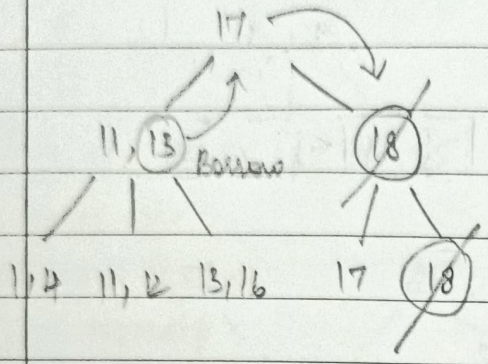
(ii) Delete 19 ⇒ UNDERFLOW

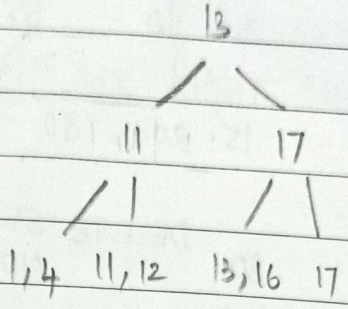
CASE (i)



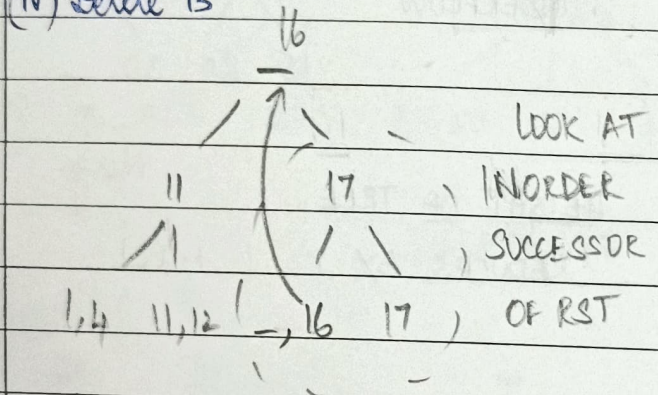
Any node can be in internal node once and leaf node

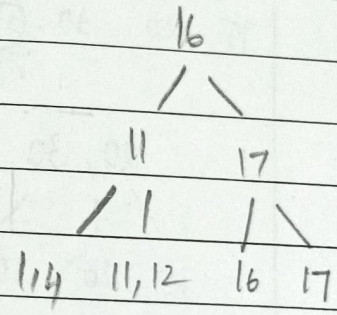
(iii) Delete 18.



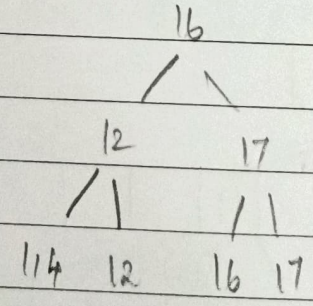


(iv) Delete 13

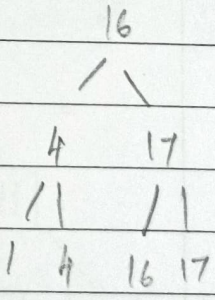




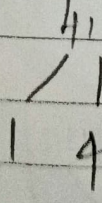
(v) Delete 11

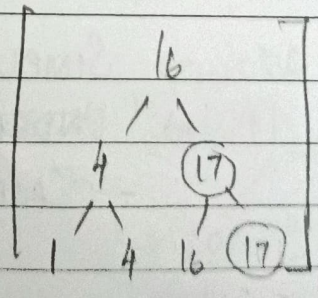


(vi) Delete 12



(vii) Delete 17

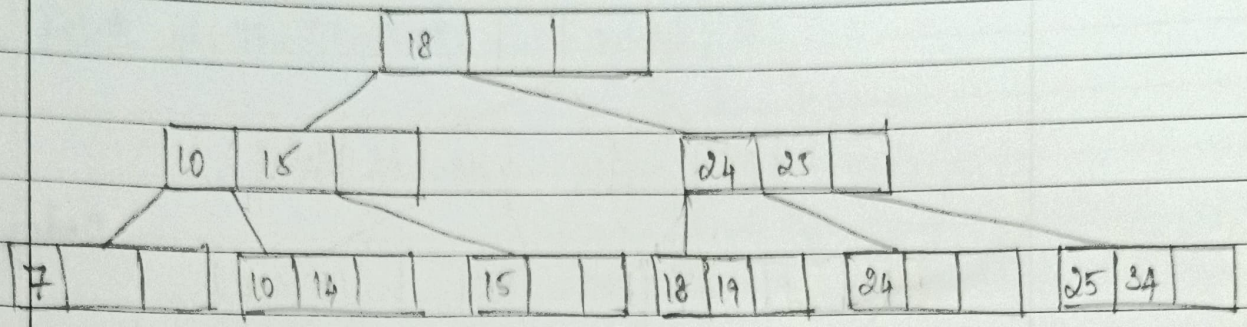




Borrow not feasible

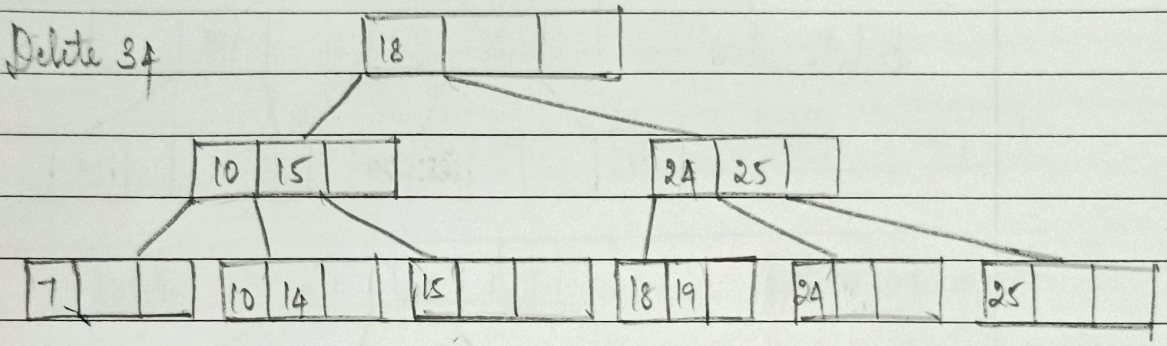
⇒ HEIGHT SHRINKS

__/__/____

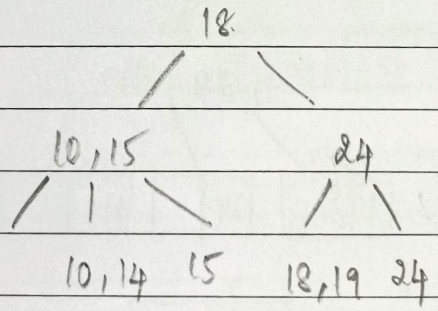


Delete 34, 25, 18, 24, 19, 7

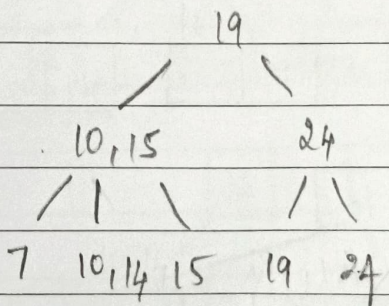
Delete 34



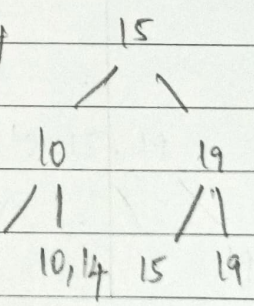
Delete 25



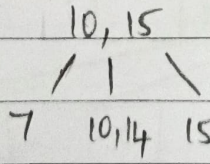
Delete 18



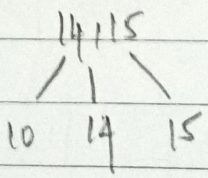
Delete 24



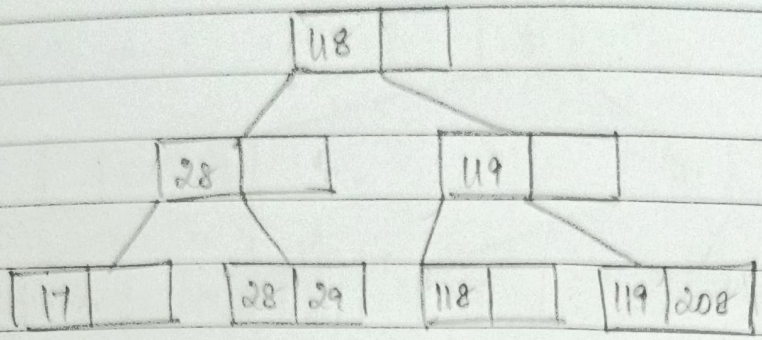
Delete 19



Delete 7



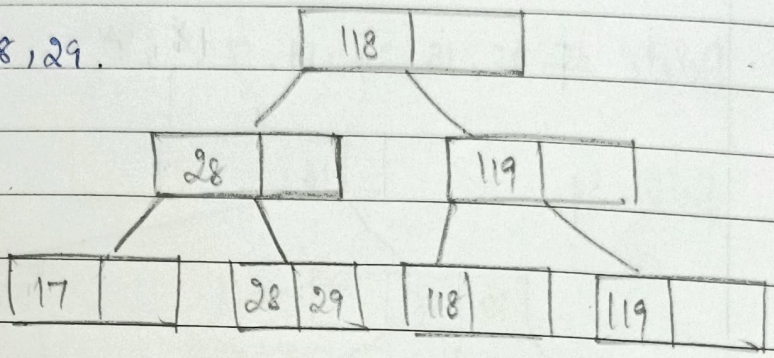
__/__/____



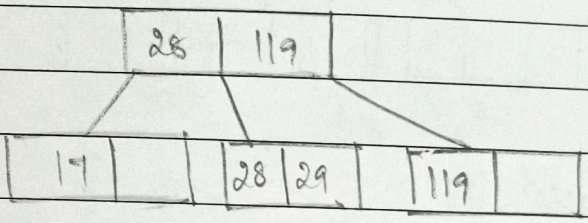
Order : 3

Delete 208, 118, 28, 29.

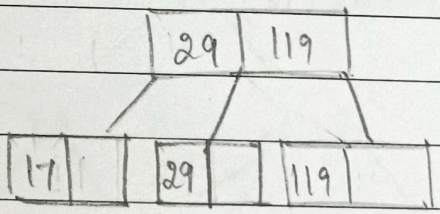
Delete 208



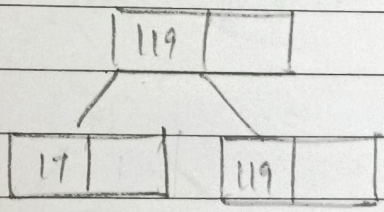
Delete 118

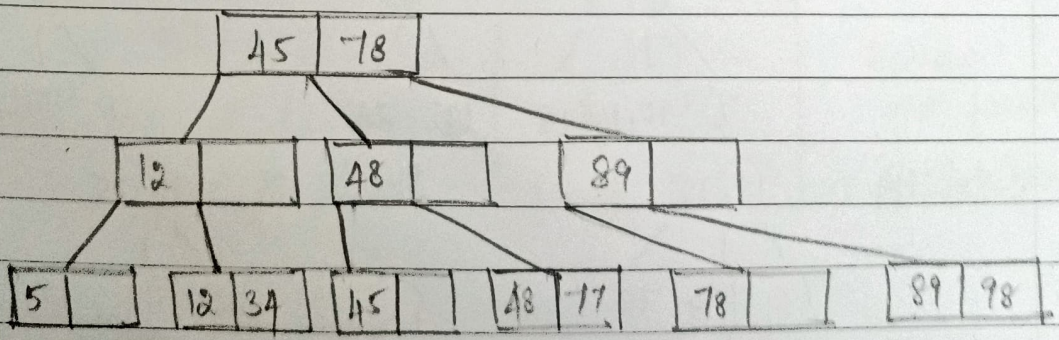


Delete 28.



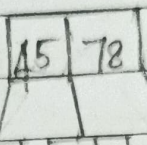
Delete 29.



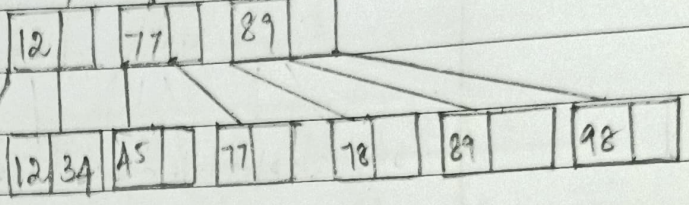


__/__/____

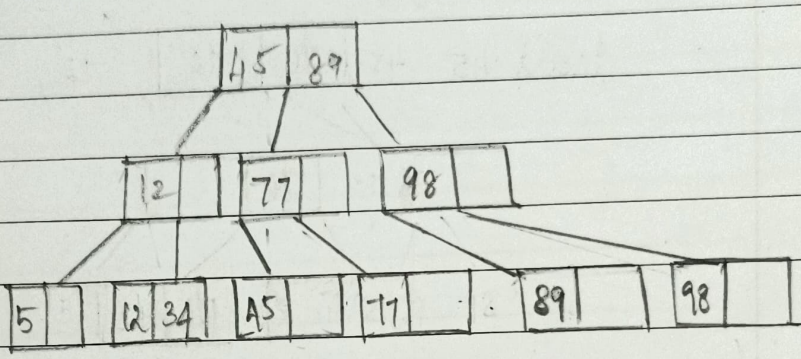
Delete 48, 78, 77, 98, 89



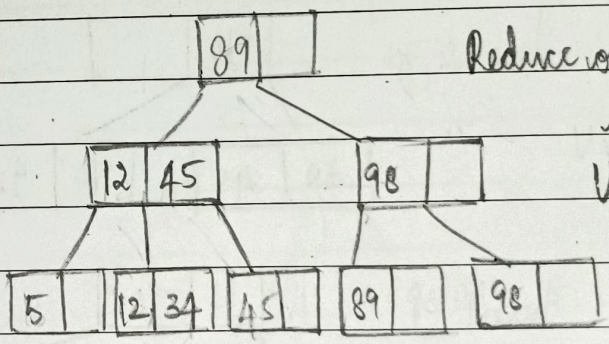
Delete 48



Delete 78

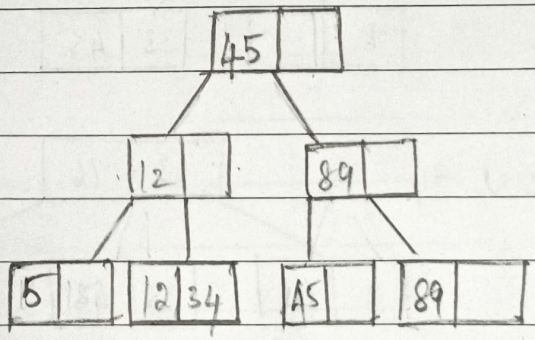


Delete 77

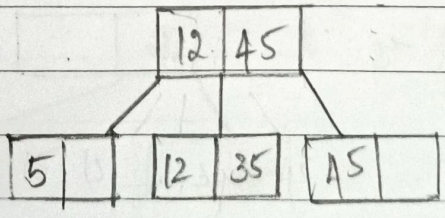


Reduce one of the key of the parent Underflow

Delete 98



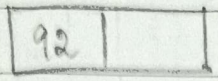
Delete 89.



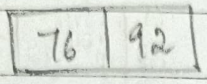
__/__/____

Insert 92, 76, 45, 33, 1, 94, 78, 85 Order $m = 3$.

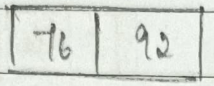
Insert 92.

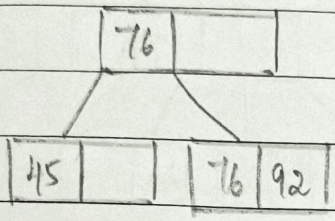


Insert 76

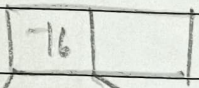


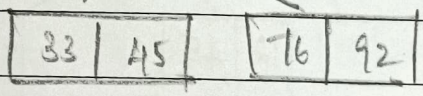
Insert 45 45



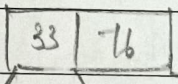


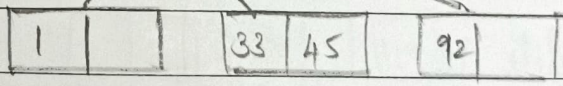
Insert 33



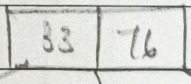


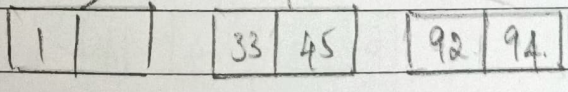
Insert 1



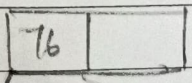


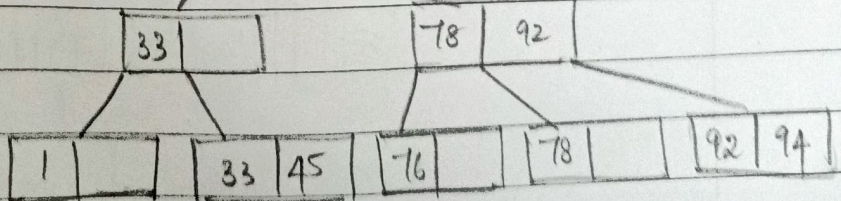
Insert 94





Insert 78





__/__/__

Insert 85