

UNIT: NO: 5: (FIVE):-

INDEXING AND MULTIWEIGHT TREES.Primary Indexing:-

A sequential order file whose index is ordered is

Dense Index:-

For every search value in index file or data file, there is an index record.

Index File / Data File

ID	INDEX		ID	NAME	ADDRESS	MARKS
101		→	101	A		
102		→	102	B		
103		→	103	C		
104		→	104	D		
105		→	105	E		

NOTE:- INDEXING:-

Two categories: (1) Primary Indexing

— (1) Dense Indexes

— (2) Sparse Indexes.

(2) Secondary Indexing.

Sparse Index:-

- reduces index file

- index record appears only for few data/items in the index/data file.

⇒

ID	INDEX	ID	NAME	ADDRESS	MARKS
101		101	A		
102		102	B		
103		103	C		
		104	D		
		105	E		
		106	F		

101 < 102 < 103

SECONDARY INDEXING:-

- used when size of file is very large.
- Two level indexing is another name
- It is used to create additional indexes of data storing the file.
- Improves perf of complex queries and simplifies search time
- It can be created or dropped at any time

ID		ID	NAME	ADDRESS	MARKS
1		1	A		
101		2	B		
201		3	C		
⋮		⋮	D		
		101	E		
		102	F		

201

* TREE (INDEXING)

(1) B Tree.

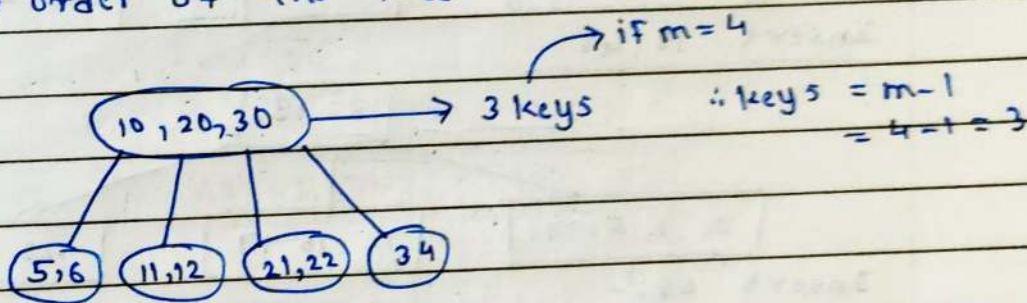
B tree of order 'm' is a multilevel tree/m-way tree in which all leaves (leaf nodes) are at the same level.

All nodes except the root node have the keys between $(m-1)/2$ and $m-1$.

The root node is either a leaf node or it has 2 to m children

The major application of B tree is Indexing.

Here, $m \rightarrow$ Order of the Tree.

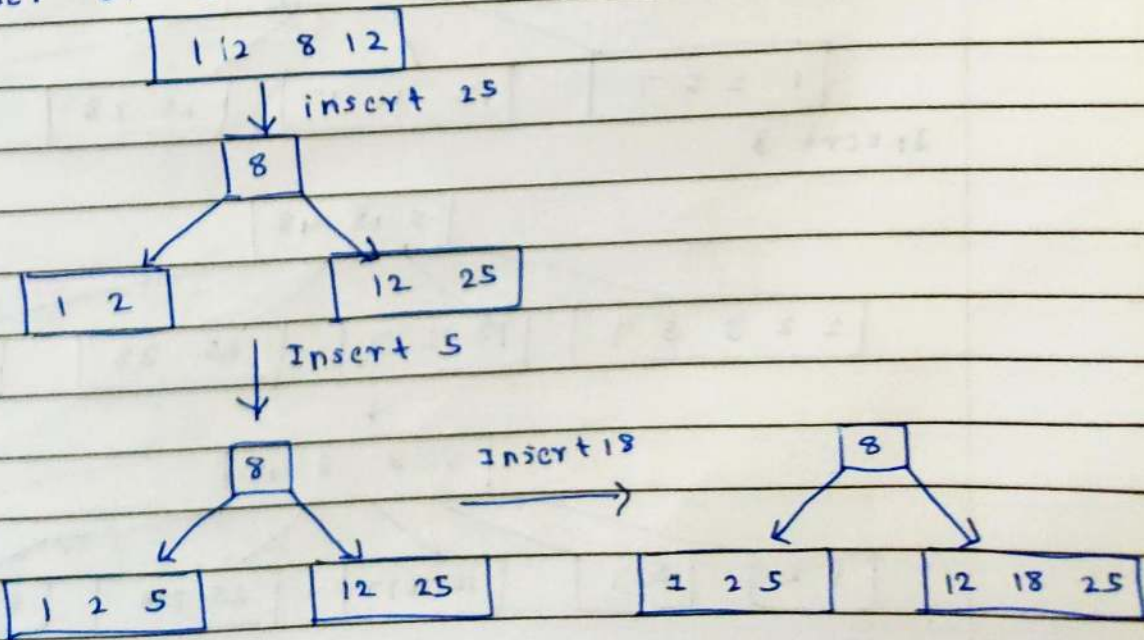


CREATION:-

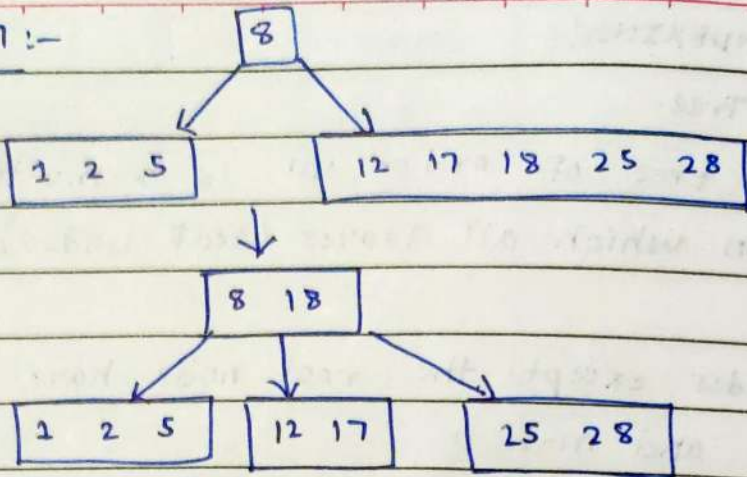
Suppose we start with an empty B Tree and keys arrive in the following order.

1, 8, 2, 8, 12, 25, 5, 18, 28, 17, 7, 52, 16, 48, 68, 3, 26, 29, 53, 55,

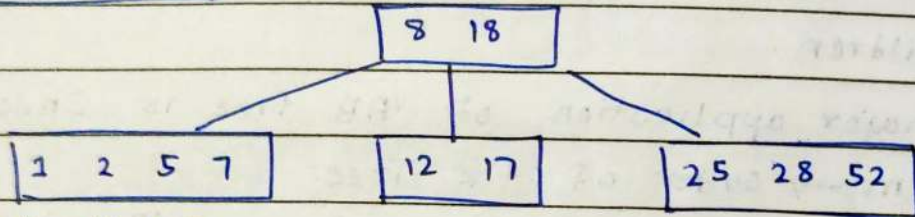
45. Order of B Tree (m) = 5



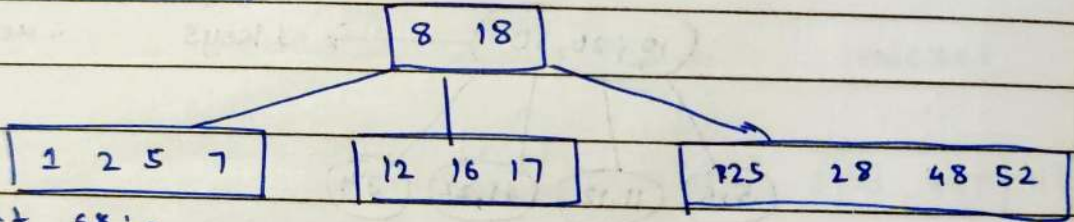
Insert 28, 17 :-



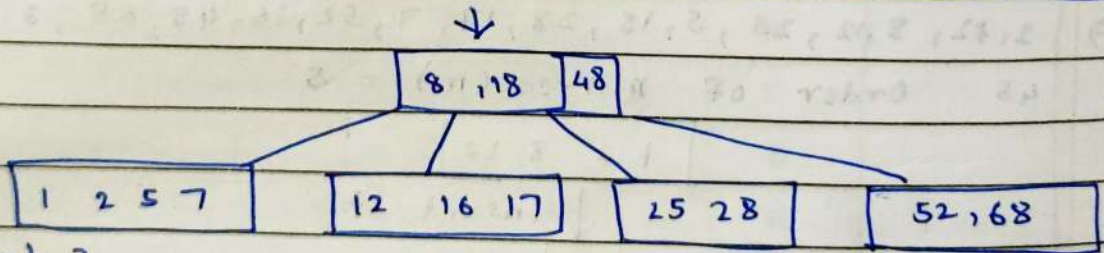
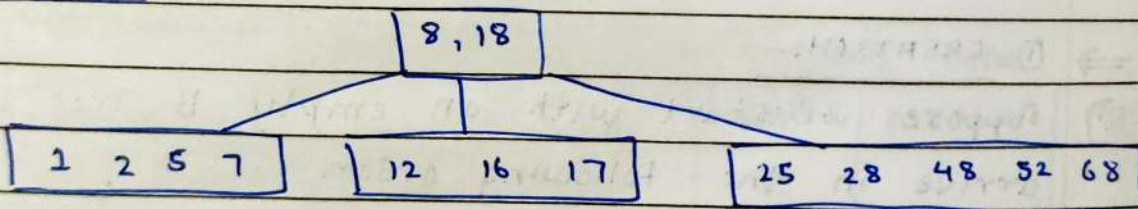
Insert 7, 52 :-



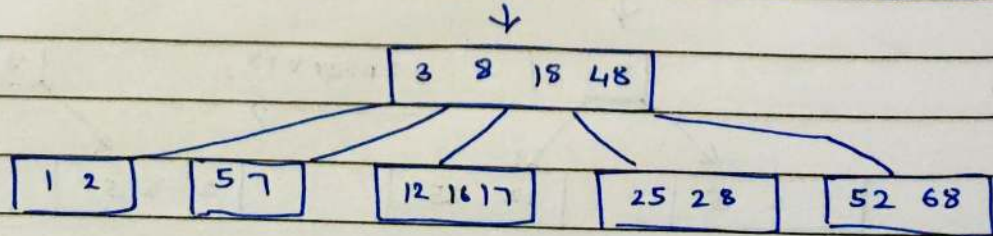
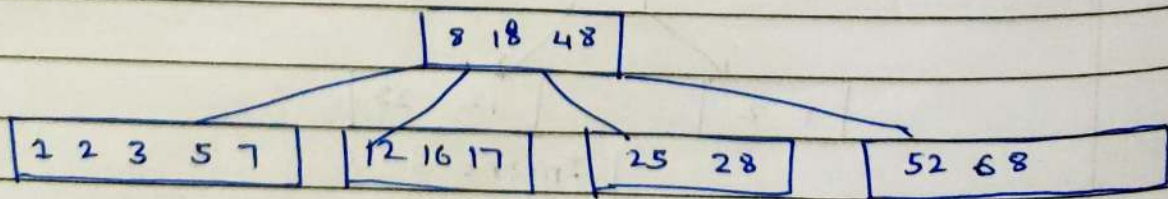
Insert 16, 48 :-

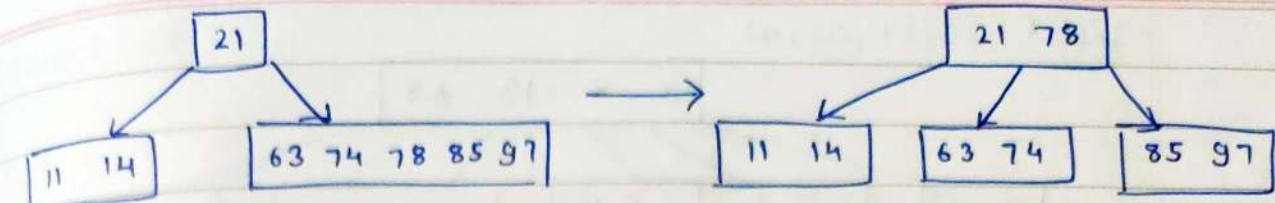


Insert 68 :-

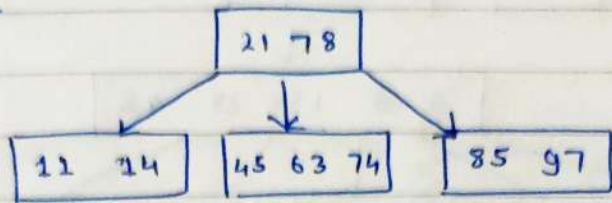


Insert 3

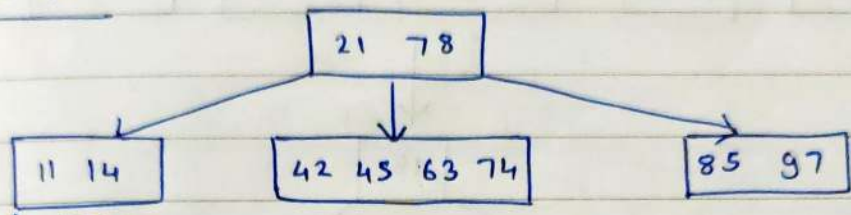




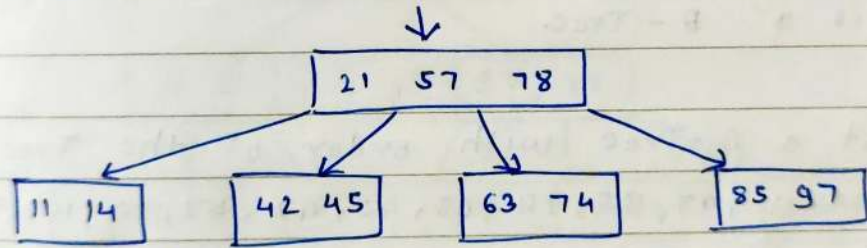
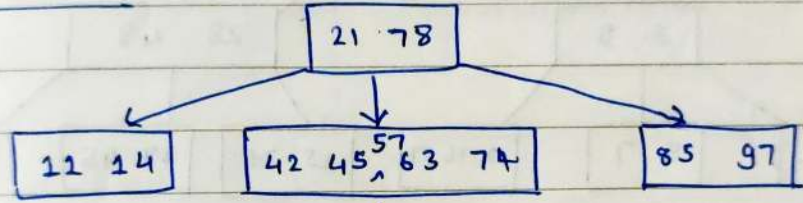
→ Insert 45:-



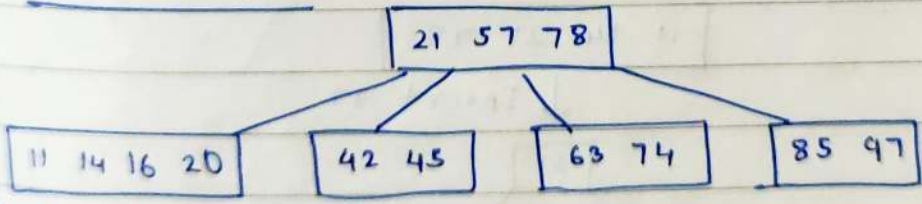
→ Insert 42:-



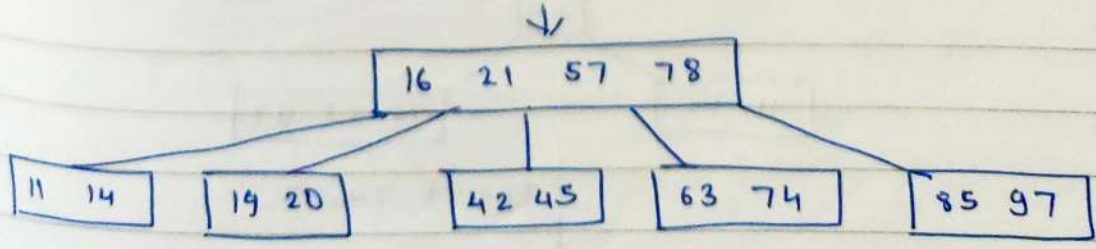
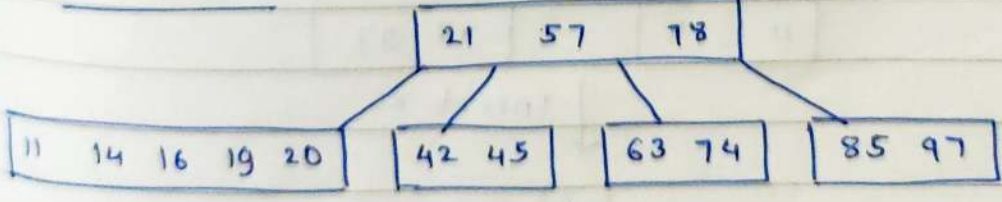
→ Insert 57:-



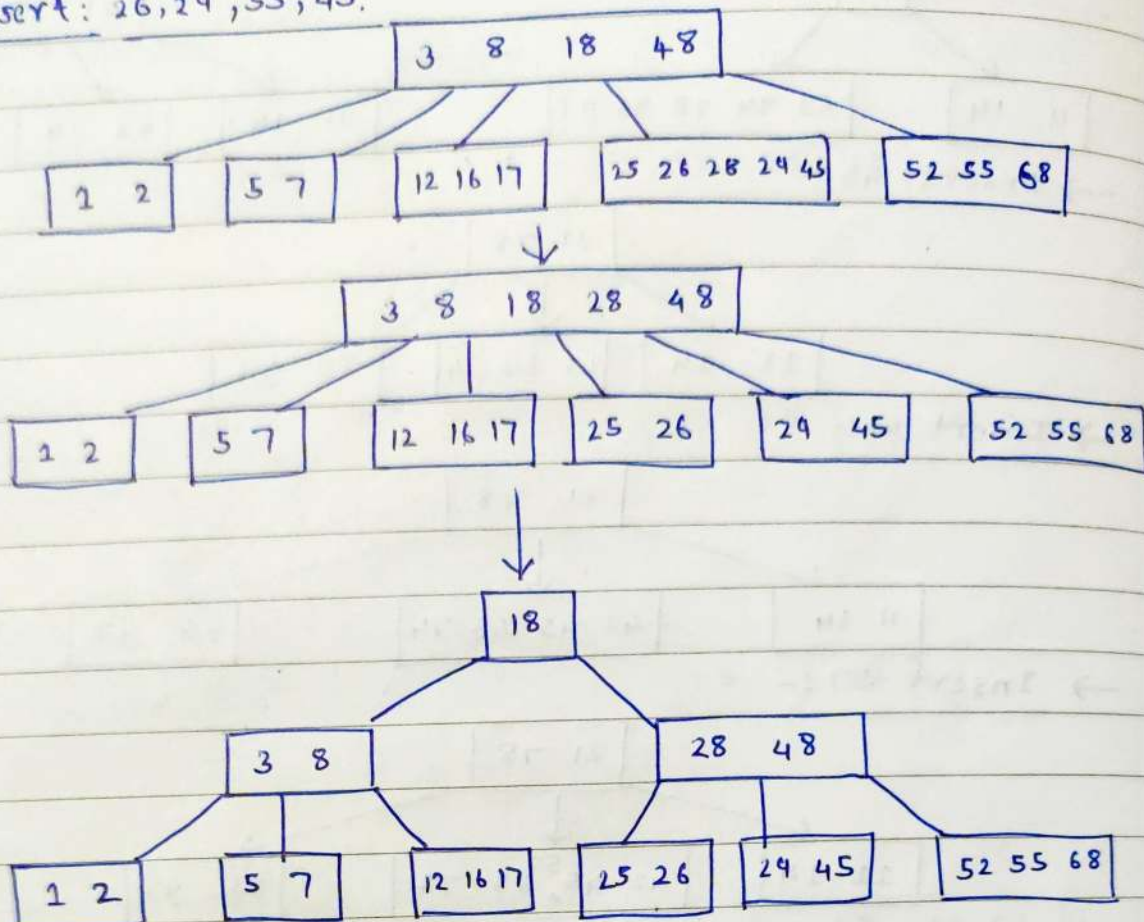
→ Insert 20, 16:-



→ Insert 19:-



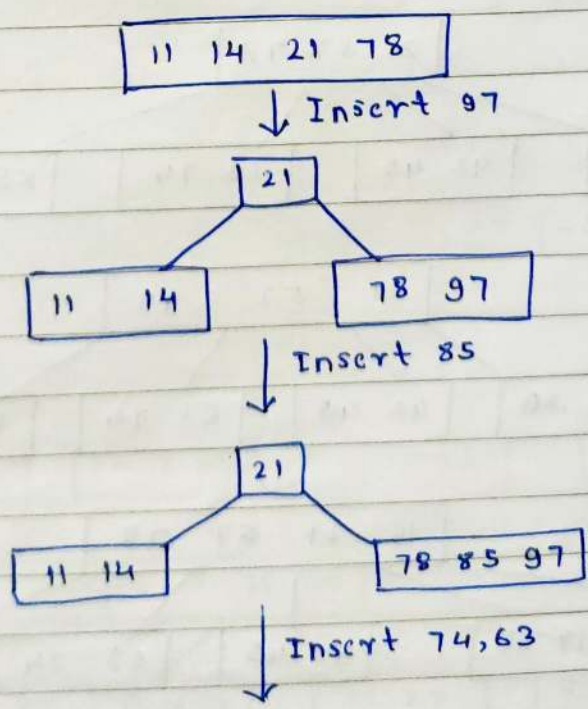
Insert: 26, 24, 55, 45.



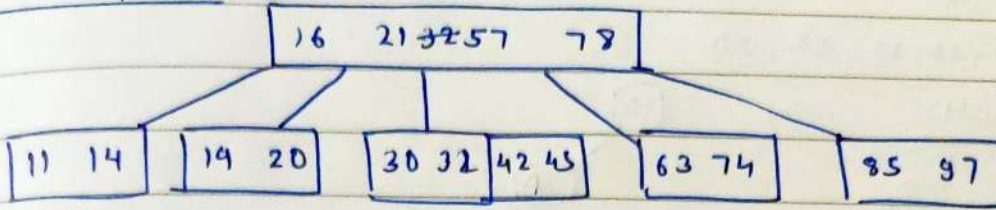
• This is a B-Tree.

Q] Construct a B-Tree with order of the Tree. $m=5$
 → 78, 21, 14, 11, 97, 85, 74, 63, 45, 42, 57, 20, 16, 19, 32, 30, 31

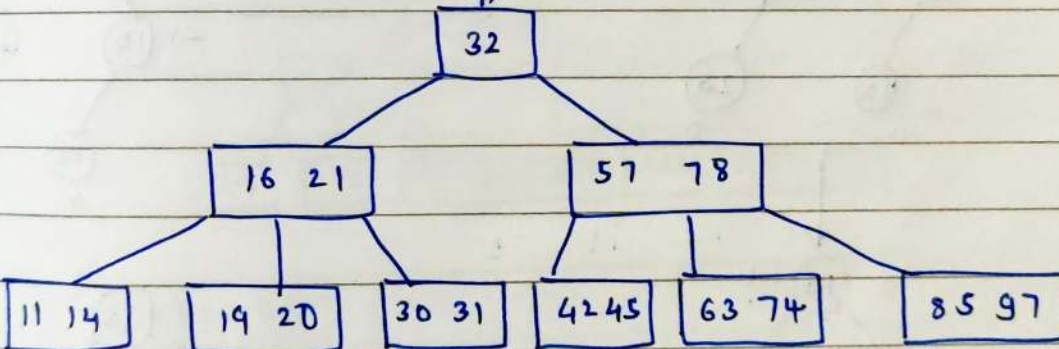
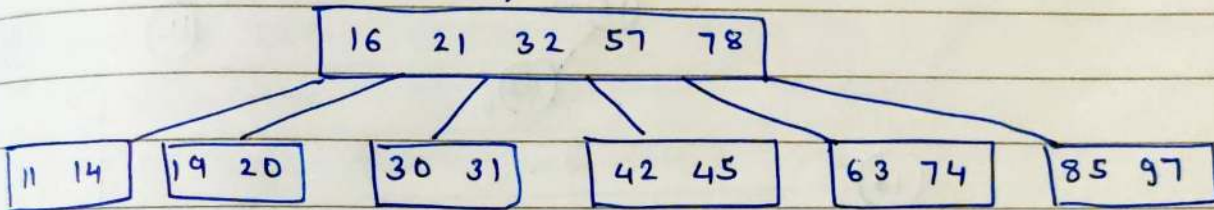
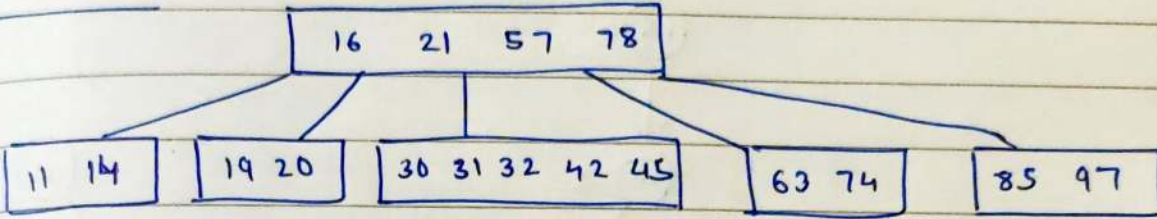
ANS. Solution:-



Insert 32, 30 :-



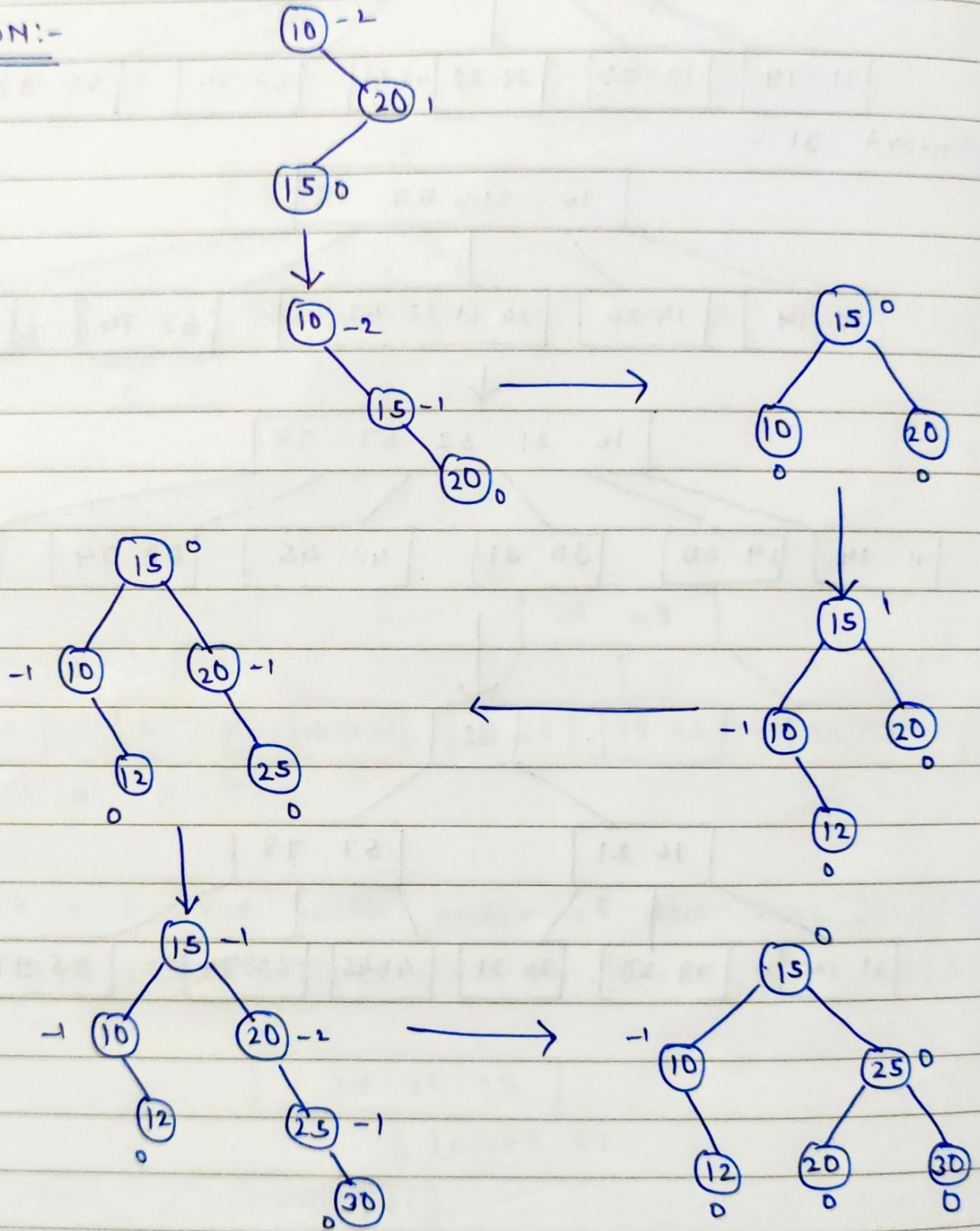
Insert 31 :-



Q. Create an AVL Tree for the following data by insert.

10, 20, 15, 12, 25, 30

ANS. SOLUTION:-



This is an AVL Tree

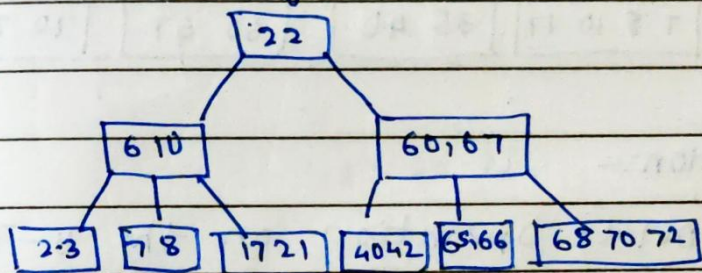
DELETION:-

If the key is already the root node, simply delete that key.

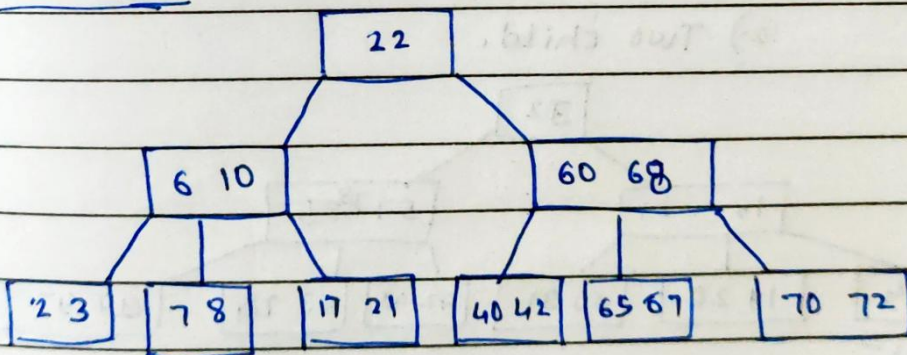
If deletion of the key causes the node to have $(m-1)/2$ keys then underflow occurs. In this case, if the left neighbour contains more than $(m-1)/2$ keys, then it can borrow key from left neighbour.

If the right neighbour contains more than $(m-1)/2$ keys, then it can borrow key from its right neighbour.

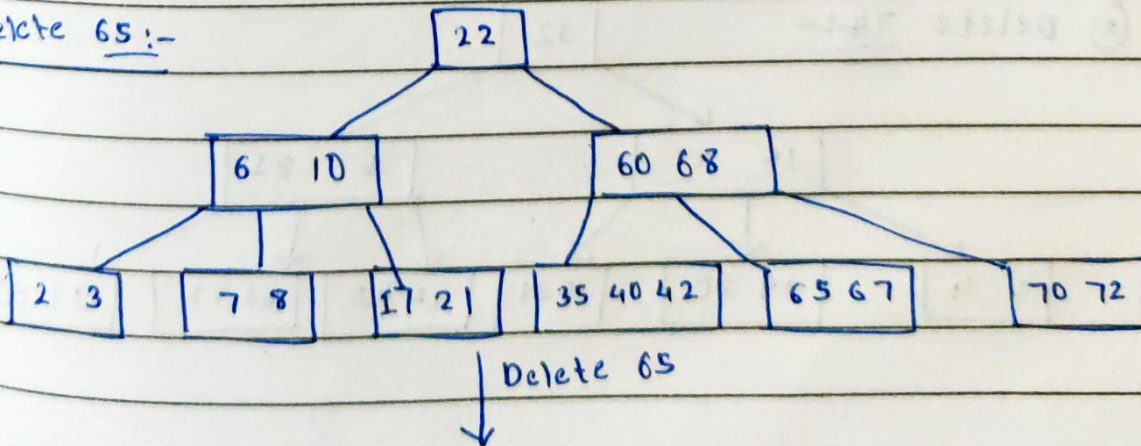
If both the left and right neighbours contain exactly $(m-1)/2$ keys than, we join together the current node and its neighbour

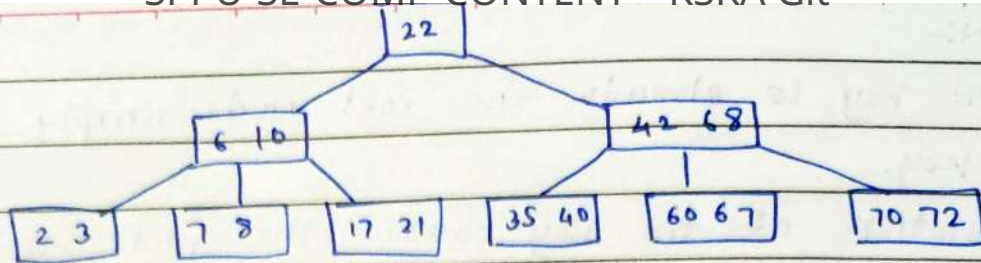


• Delete 66

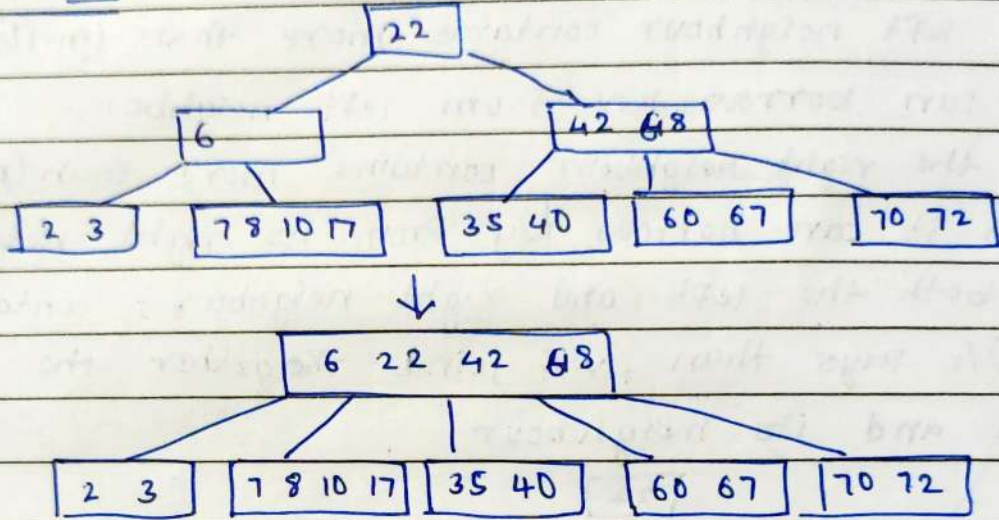


• Delete 65:-





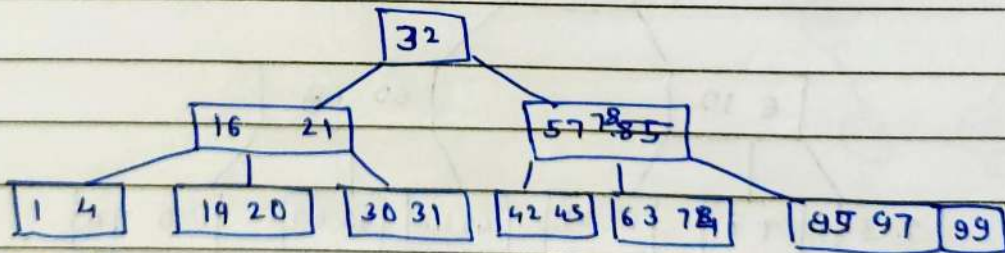
Delete 21 :-



o University Question:-

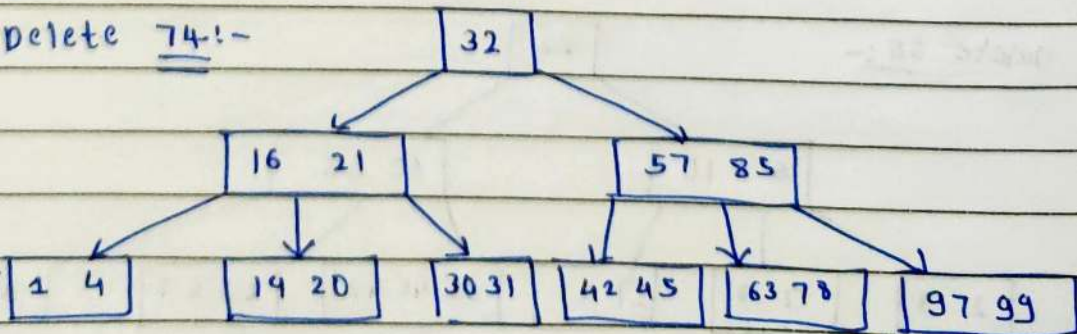
Q] Explain the delete operation on the B-Tree of the order $(m) = 5$

[Two cases:- 1) Leaf node and Underflow condition.
2) Two child.]

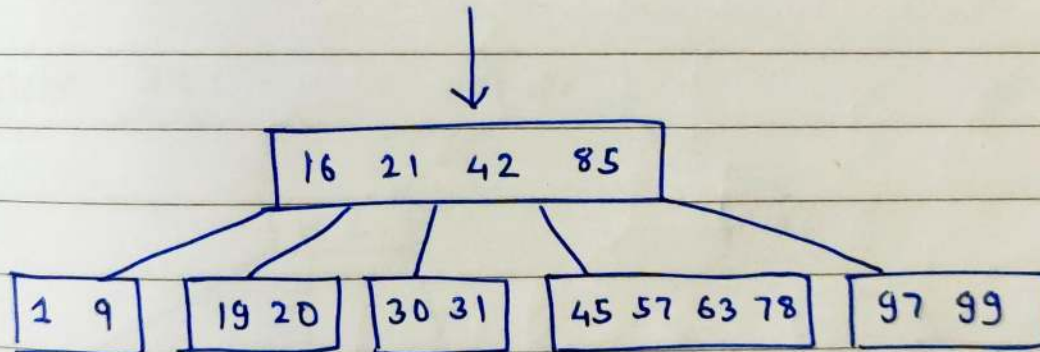
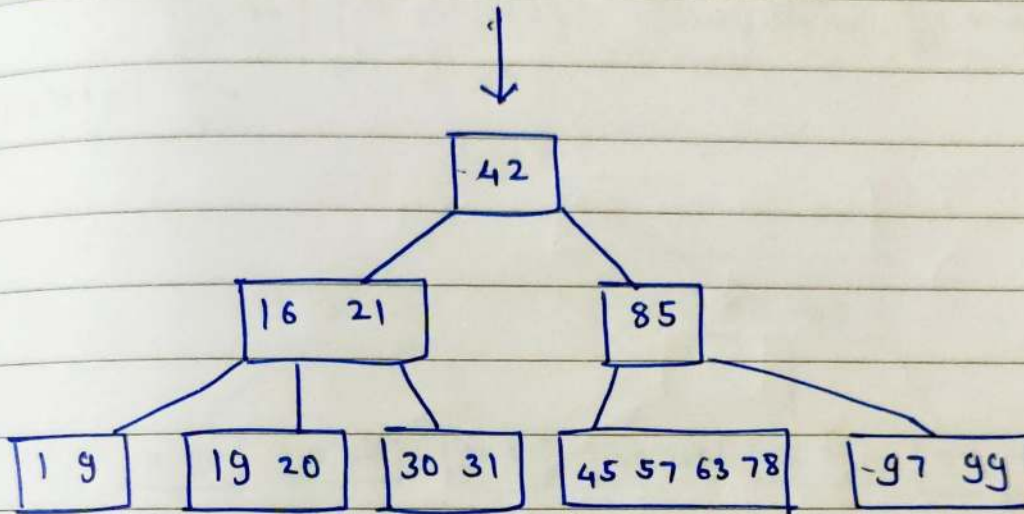
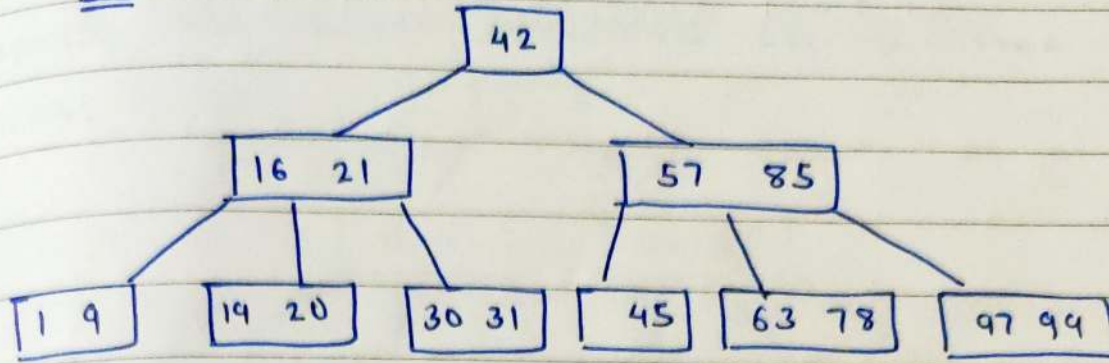


ANS. solution :-

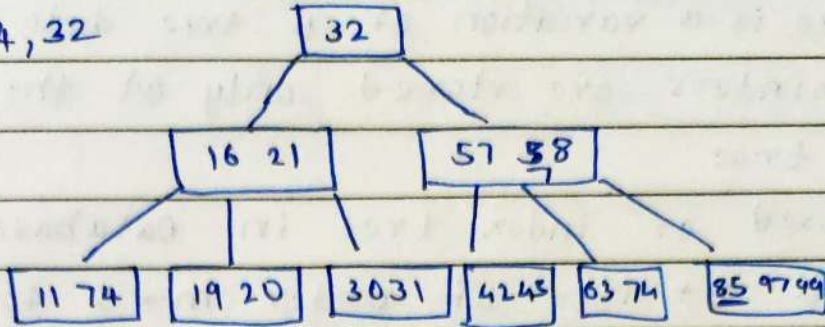
① Delete 74 :-



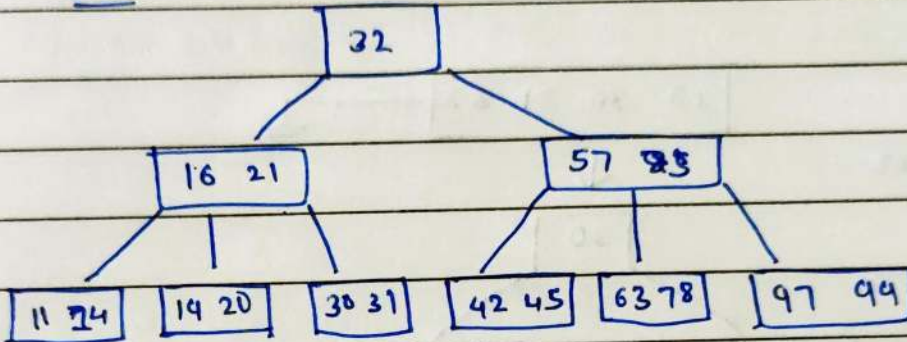
① Delete 32 (Inorder Successor.)
→



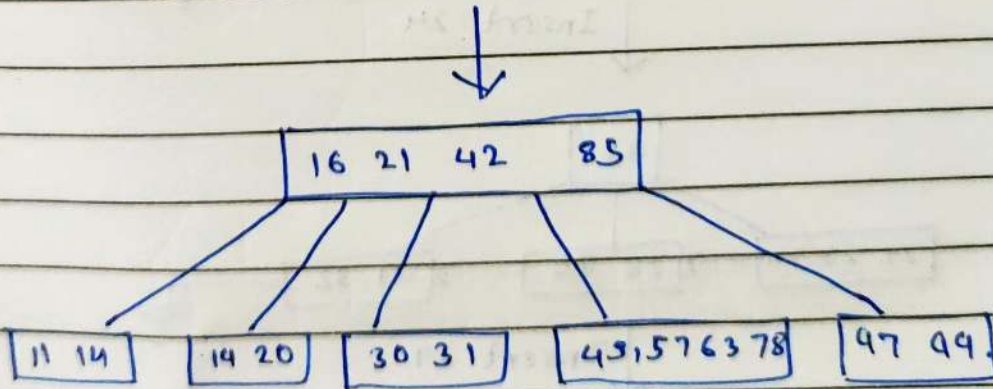
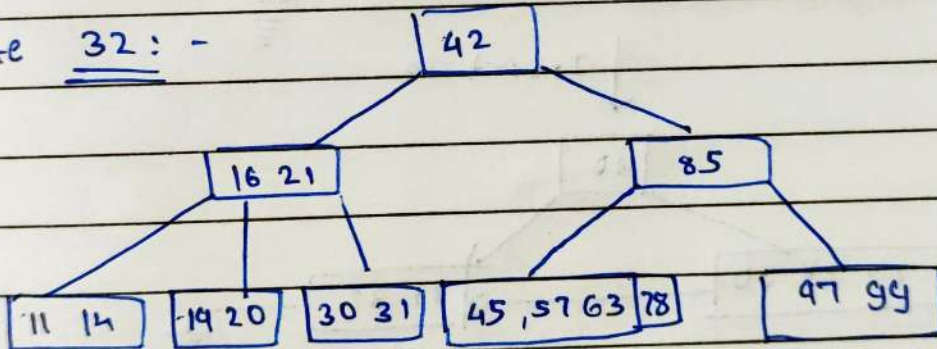
Q] Explain the delete operation on B Tree of Order 5
Delete 74, 32



Ans. Delete 74



Delete 32 :-

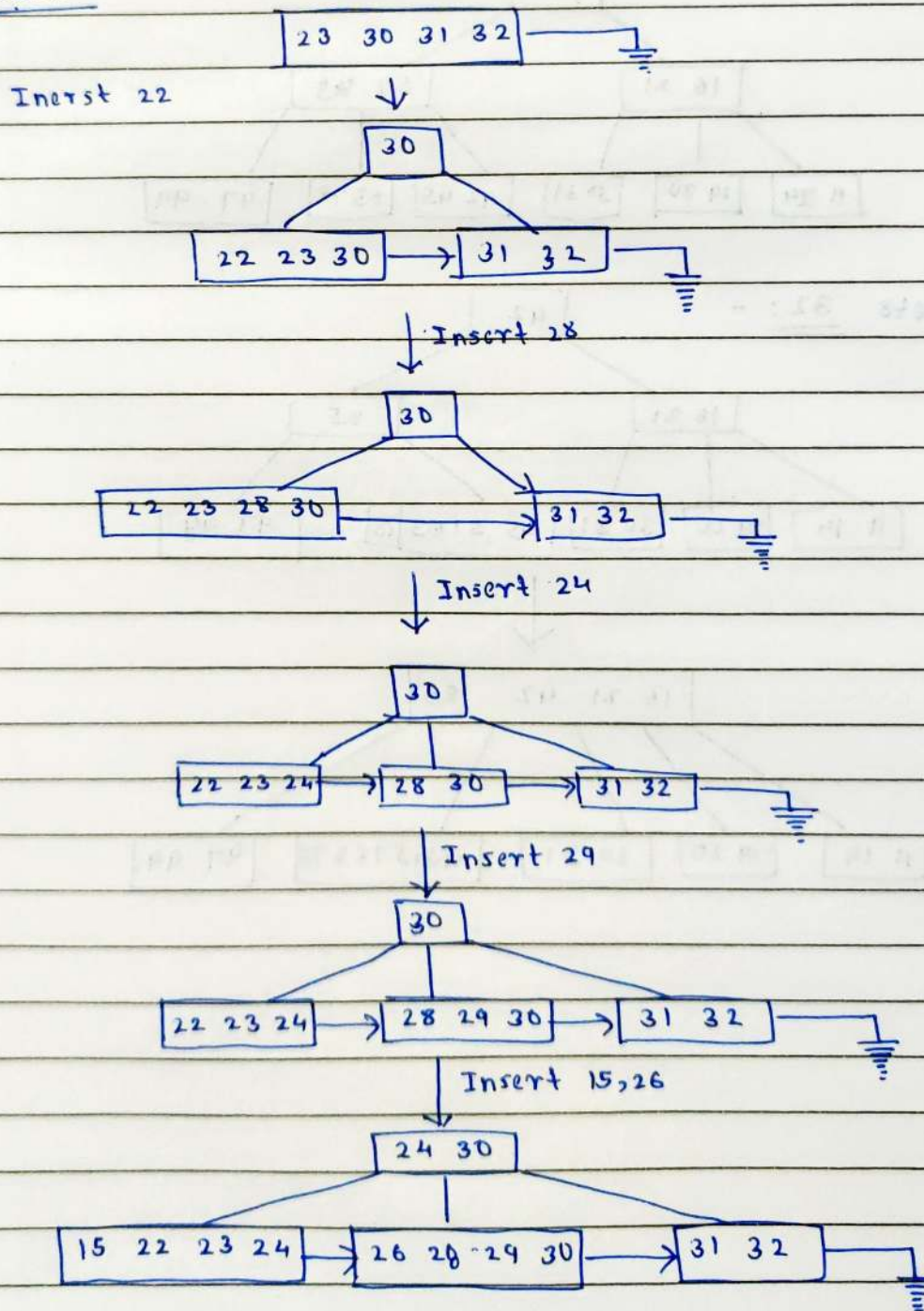


B+ Trees (Deletion not asked in Exam).

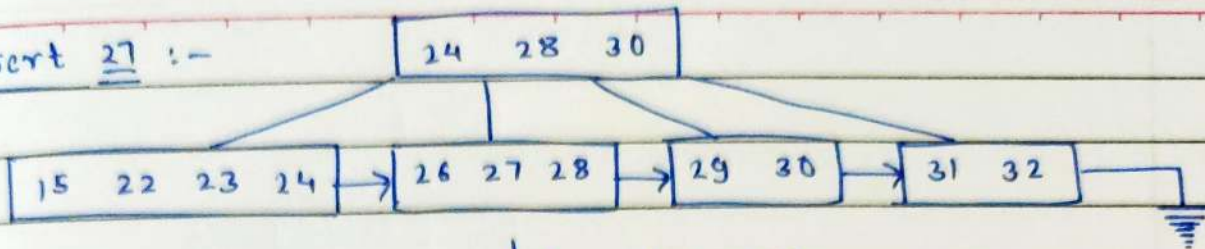
- B+ Tree is a variation of B-tree data structure.
- Data pointers are stored only at the leaf nodes of the tree
- It is used as index tree in Database Management.

Q.] Construct a B+ Tree of order $m=5$ for the following data. 30, 31, 23, 32, 22, 28, 24, 29, 15, 26, 27, 34, 39, 36.

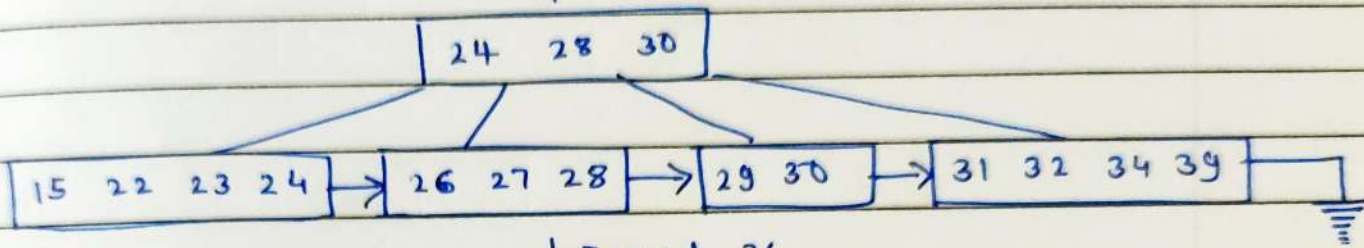
Ans: Soln:-



Insert 27 :-



Insert 34, 39.



Insert 36

