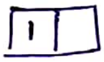
B.Tree :-

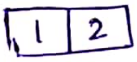
Example:- order = 3

1, 2, 3, 4, 5, 6, 7, 8, 9, 10

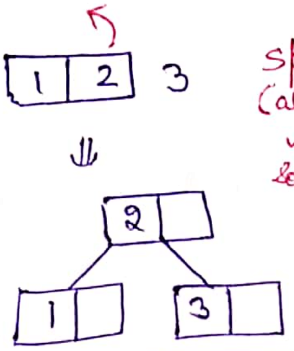
Insert 1 :-



Insert 2 :-

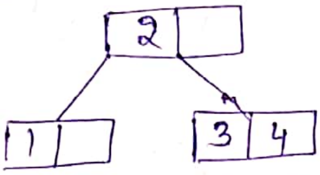


Insert 3 :-

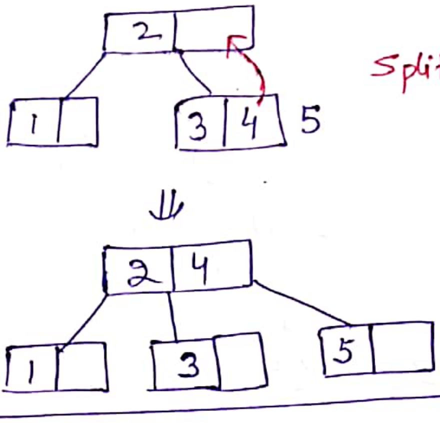


split (always at middle of sorted list)

Insert 4 :-

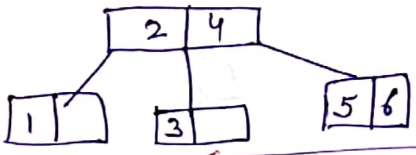


Insert 5 :-

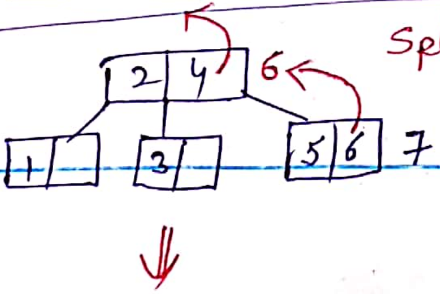


Split

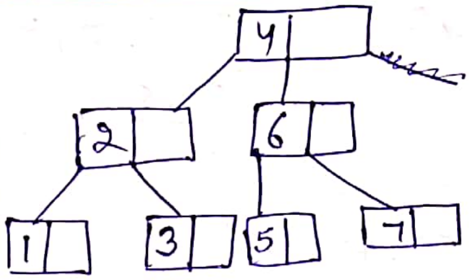
Insert 6 :-



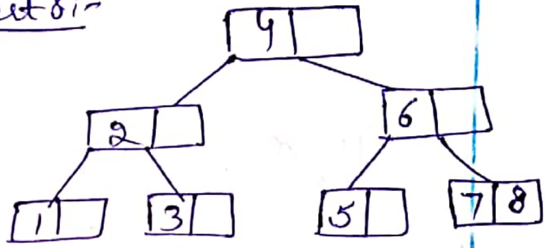
Insert 7 :-



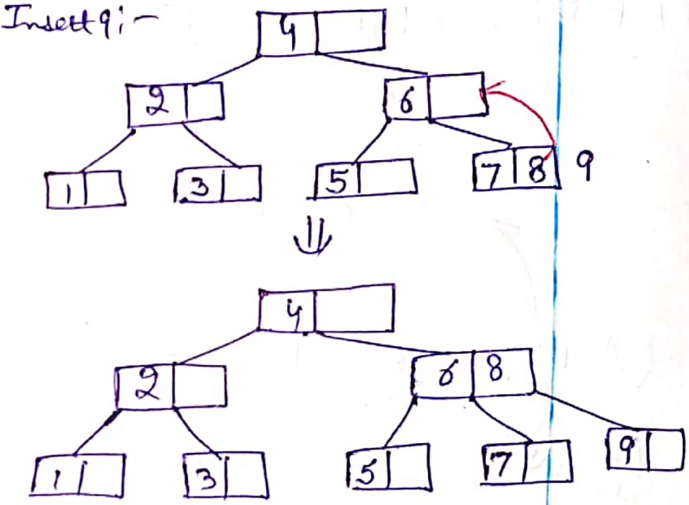
Split



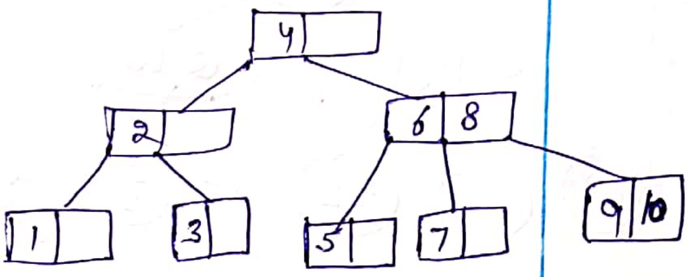
Insert 8 :-



Insert 9 :-

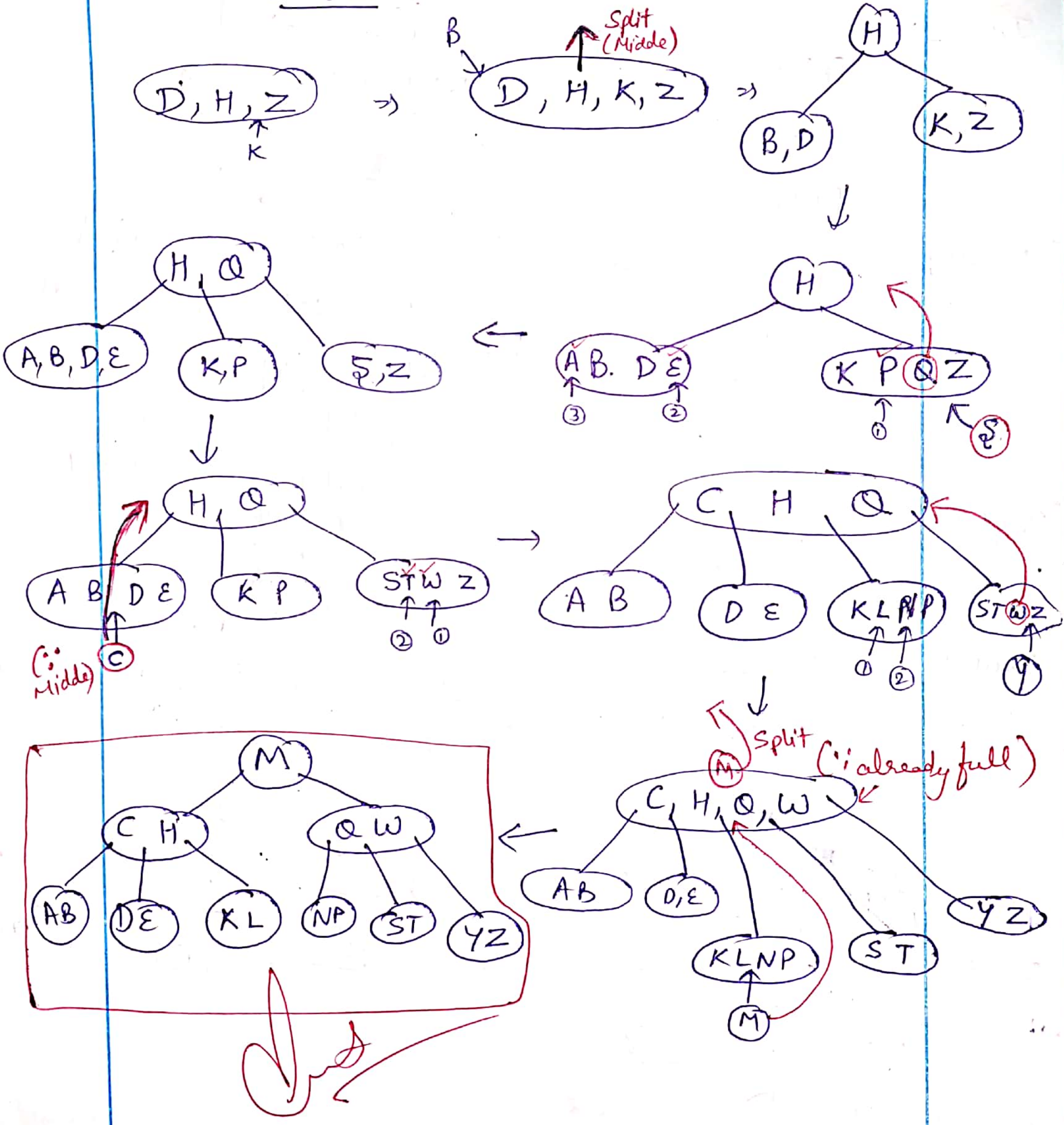


Insert 10 :-



Example 2 :- D, H, Z, K, B, P, Q, E, A, S, W, T, C, L, N, Y, M

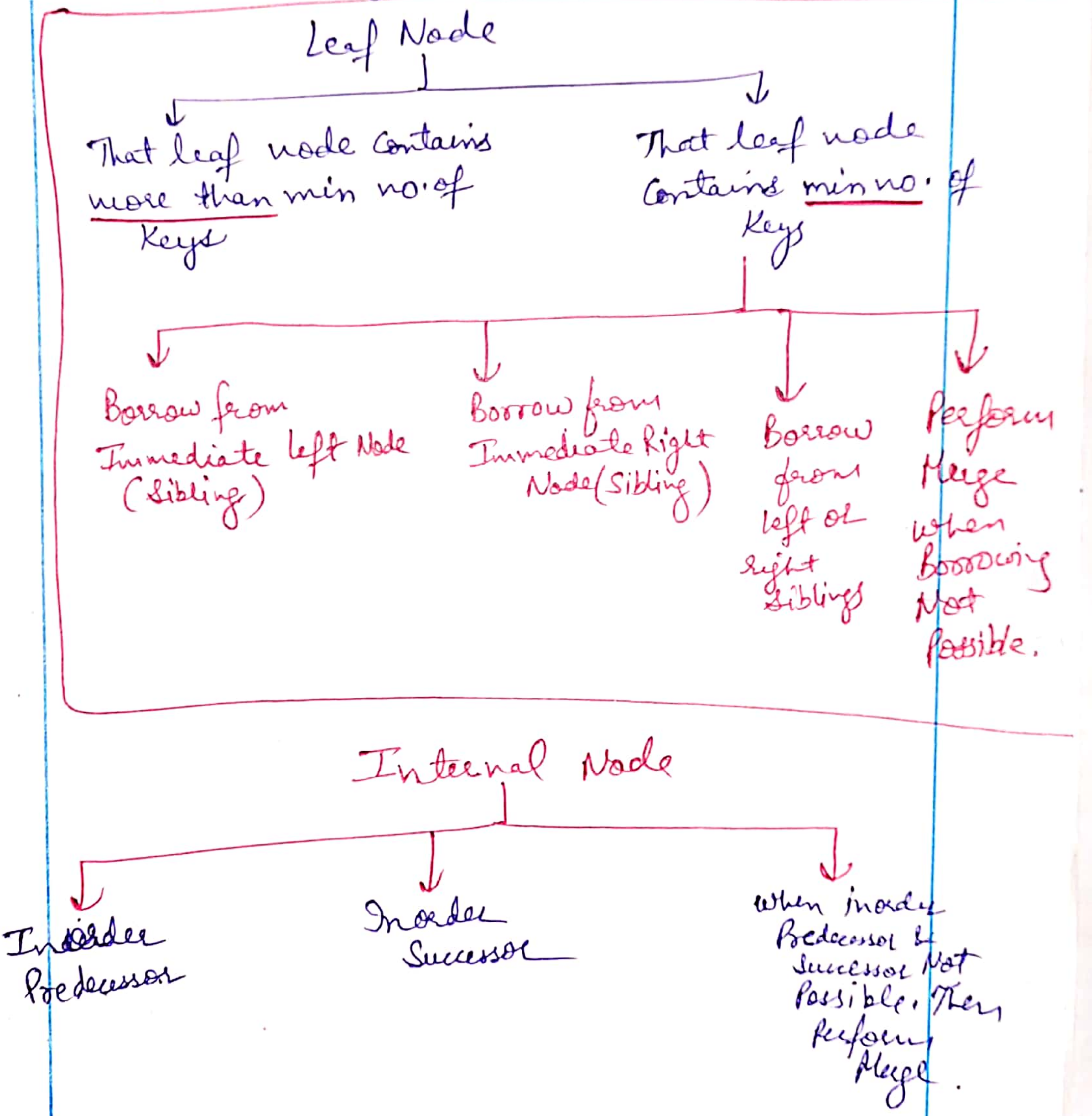
order = 5



Deletion of B-Tree

① If target key is a leaf node

② If target key is in Internal node.



Leaf Node

That leaf node contains more than min no. of keys

That leaf node contains min no. of keys

Borrow from Immediate left Node (Sibling)

Borrow from Immediate Right Node (Sibling)

Borrow from left or right siblings

Perform Merge when Borrowing Not Possible.

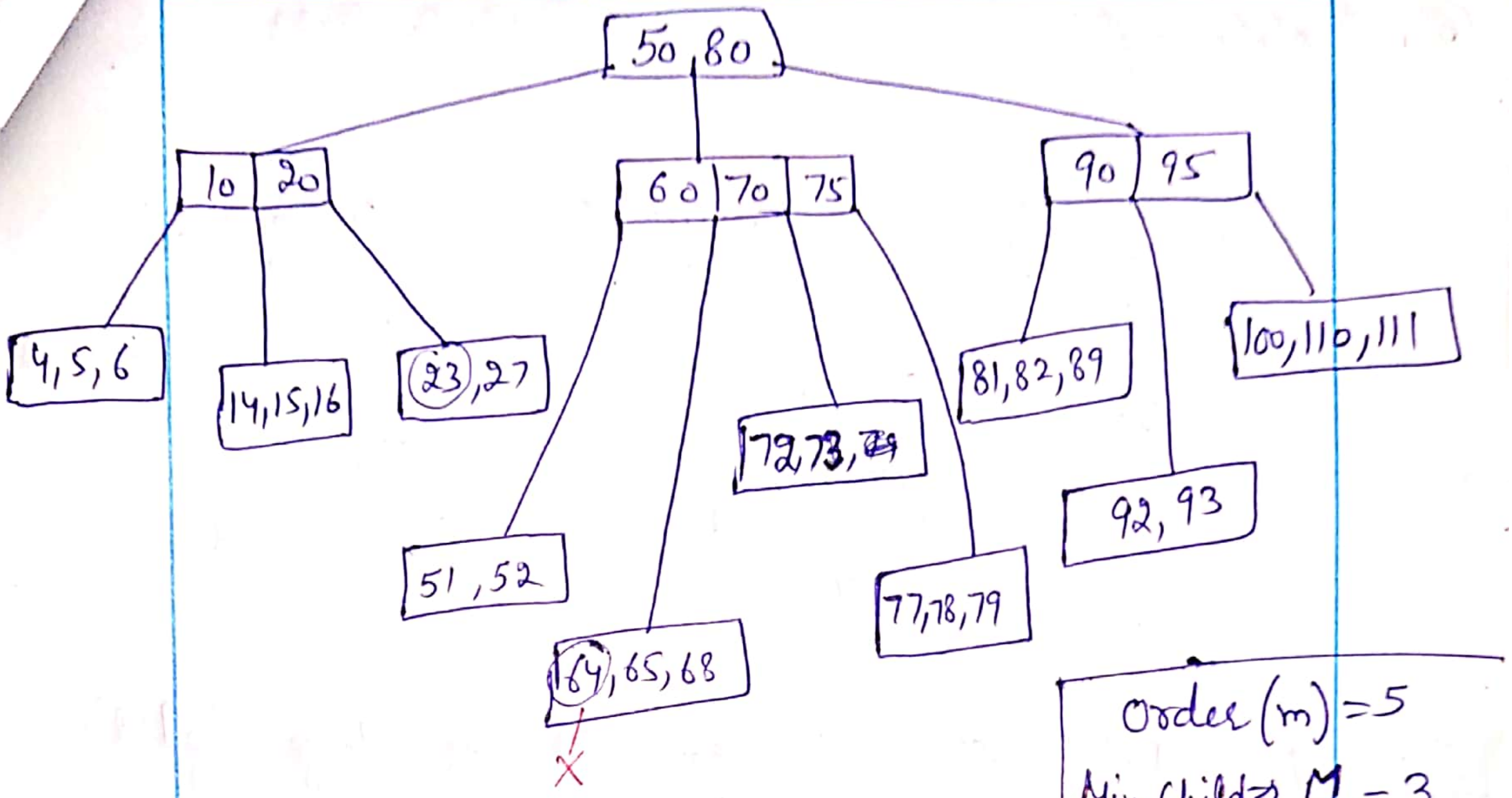
Internal Node

Inorder Predecessor

Inorder Successor

When inorder Predecessor & Successor Not Possible, Then Perform Merge.

Example:-



Order (m) = 5

Min Child ⇒ $\frac{M}{2} = 3$

Max. child ⇒ 5

Min Keys ⇒ $\left(\frac{m}{2} - 1\right)$

⇒ 2

Max. Keys = 4 (m−1)

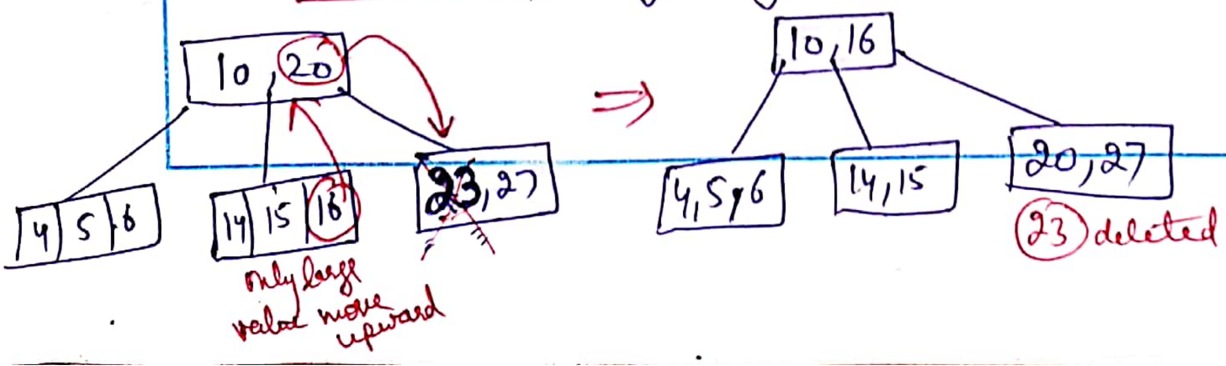
① Delete 64 :-

In this leaf node contain more than min no. of Keys, Than Simply delete 64 Key.

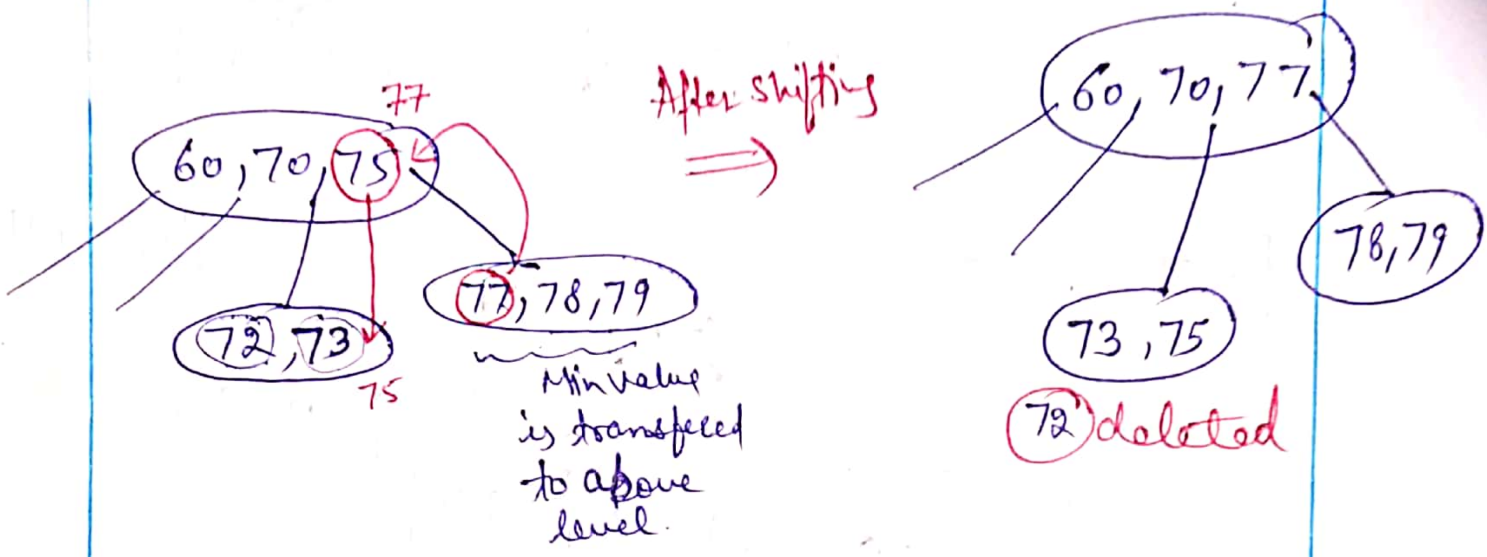
② Delete 23 :- (Borrow from Immediate left)

Leaf Node Contain min no. of Keys

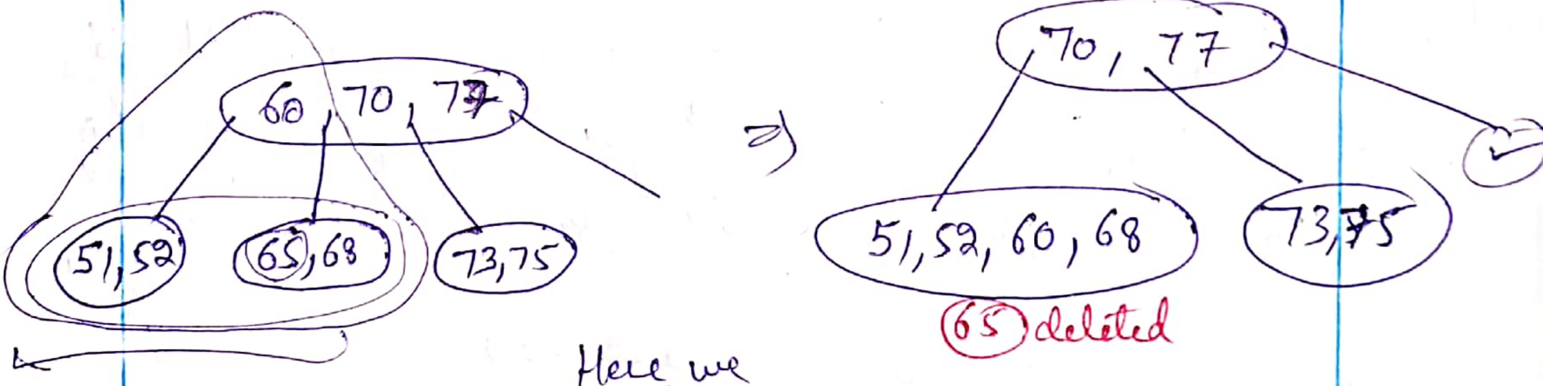
Then we can borrow key from immediate left or Right. But here no right sibling is there. So, we borrow from left sibling.. If Sibling having more than min No. of Keys. only than it will borrow.



③ Delete 72:- (Borrow from immediate Right)

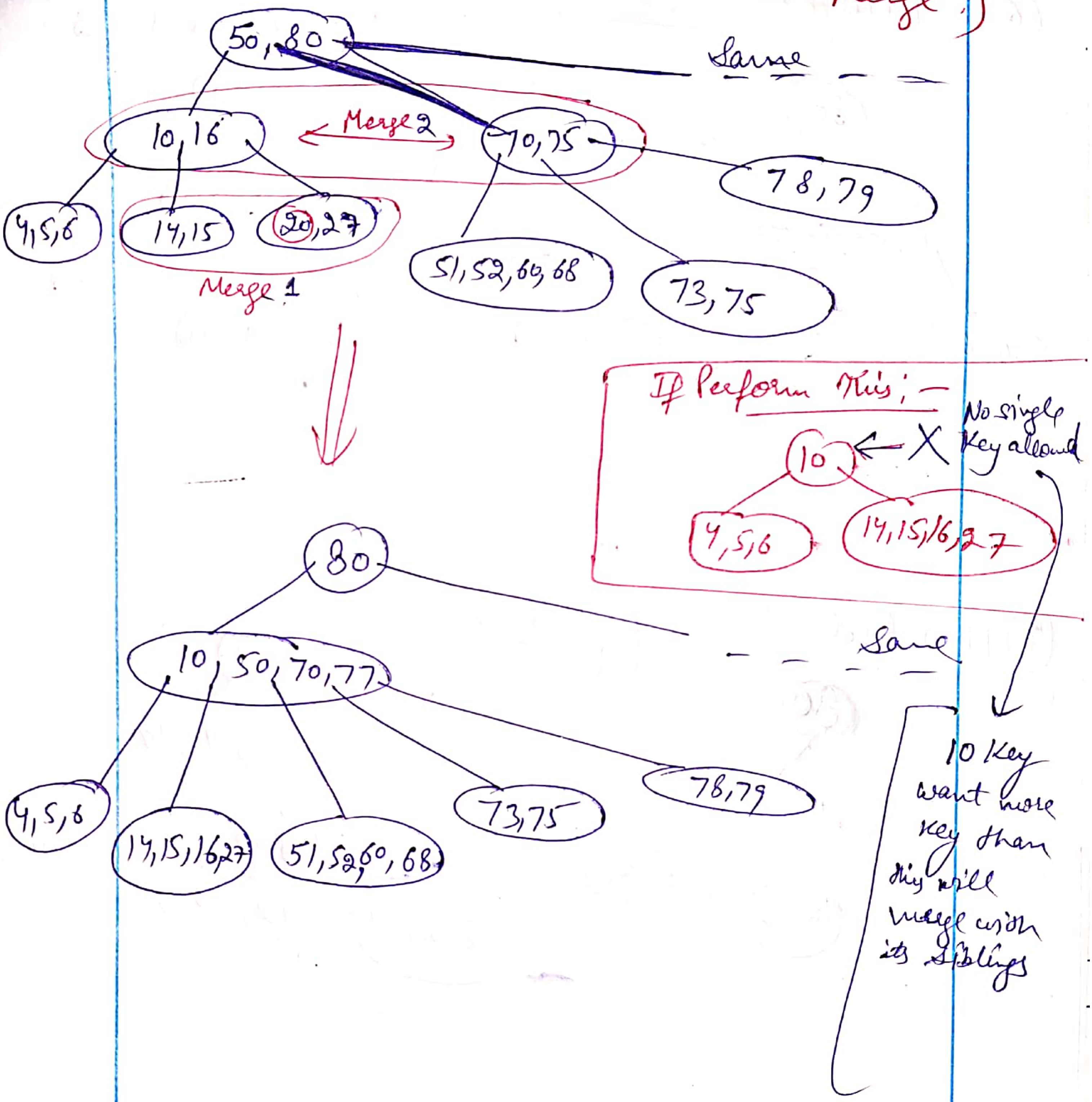


④ Now Delete 65:- (Merge left or Merge Right)



Here we merge with left Sibling or you can also merge with Right Sibling.

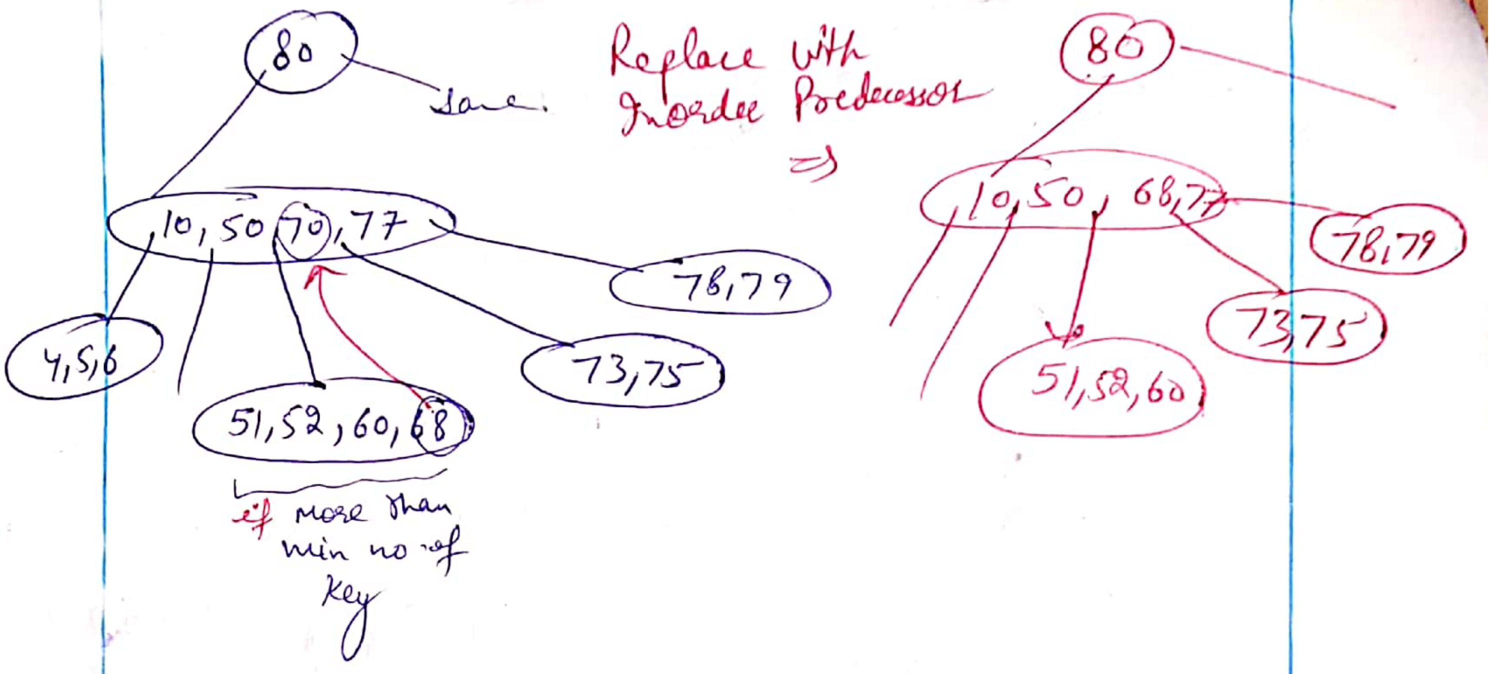
5) Now delete 20:- (when No Borrowing Possible than Merge)



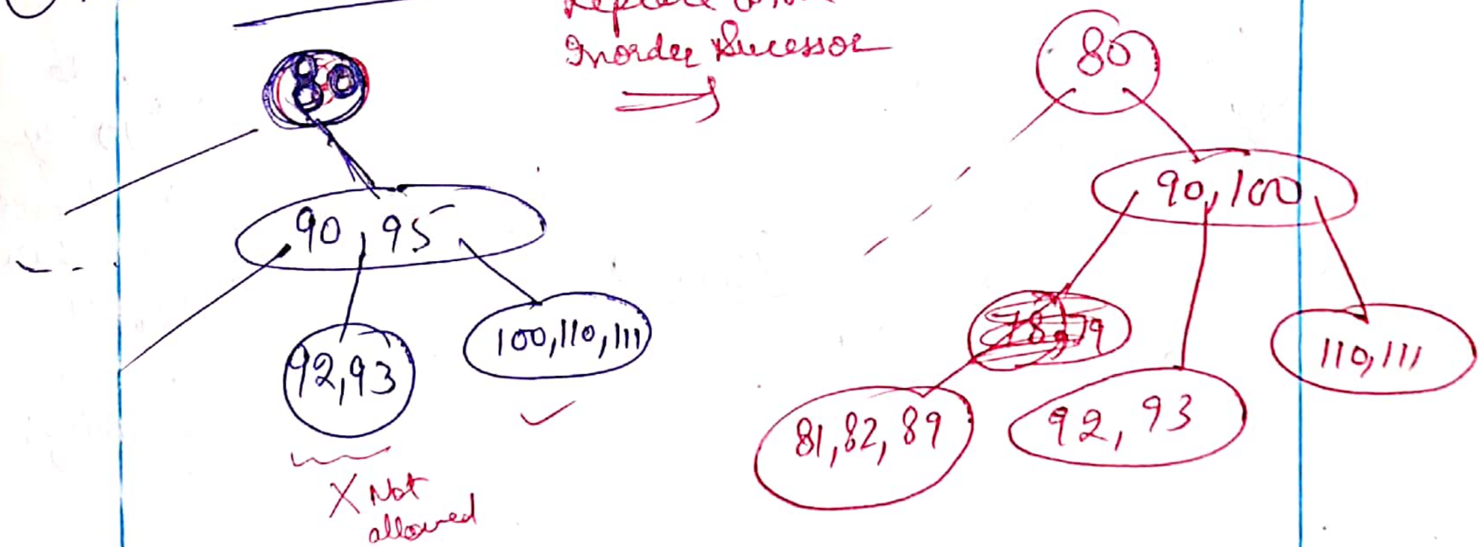
10 Key want more key than this will merge with its siblings

⑥ Now Delete 70 :- Inorder Predecessor of 70 ⇒ 68

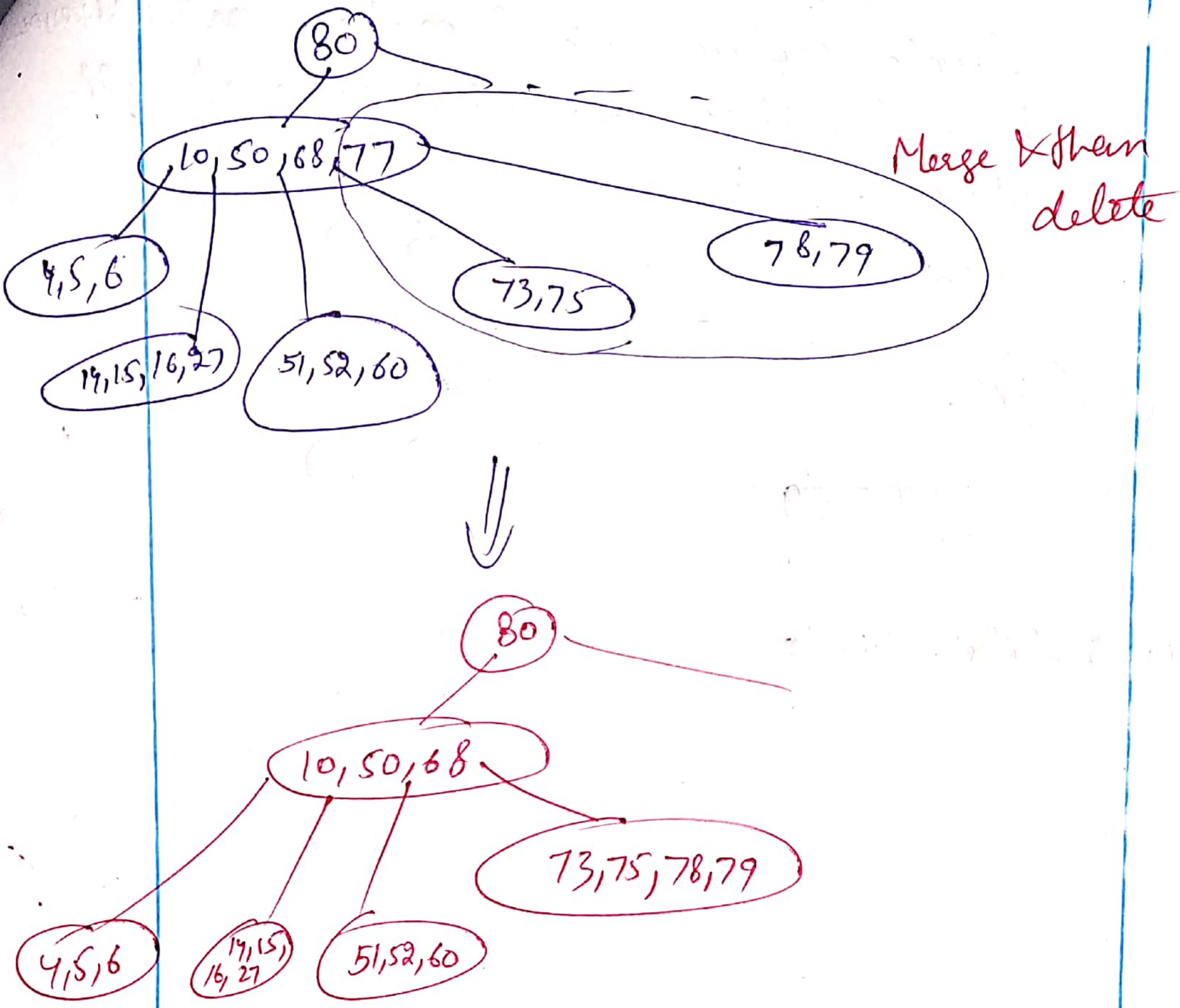
" Successor of 70 ⇒ 73



⑦ Now delete 95 :-

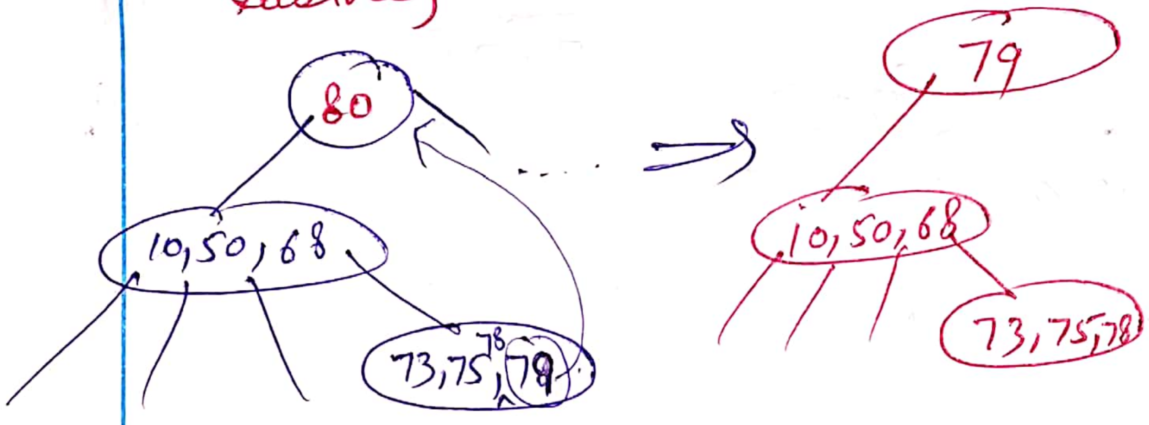


8 Now if we delete 77:-



9 Delete 80 :- (Root Node)

Simple replace with Inorder ~~[illegible]~~ Predecessor from left subtree. (i.e. Highest value of left subtree).



~~10 Delete [illegible]~~