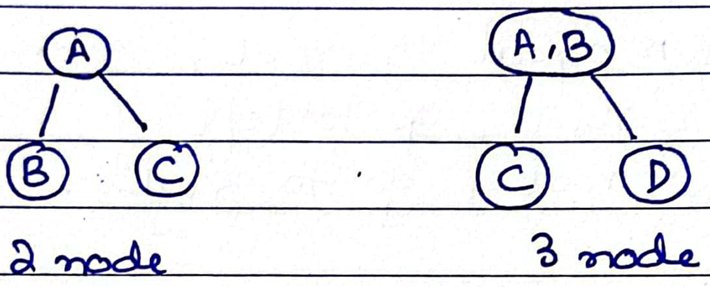
2-3 Tree.

Date:

- Height balanced tree.
- Similar to BST w/ the properties that 2-3 Tree can have ~~a~~ 2 node and 3 node.

* 2 node → 1 data/ key, 2 children.
* 3 node → 2 data/key, 3 children.



Properties :-

① Data is stored in sorted order.
② It is a balance tree.
③ All leaf nodes are on same level
④ Each node can either be leaf, 2 node and 3 node.
⑤ Insertion is always done ~~a~~ on leaf node.

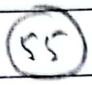
Time Complexity :-

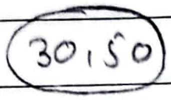
O(logn) for insertion, deletion and search.

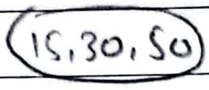
Insert in 2-3 Tree

Date:

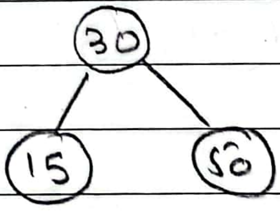
Example 01: Insert 50, 30, 15, 75, 65, 45, 5 in 2-3 Tree.

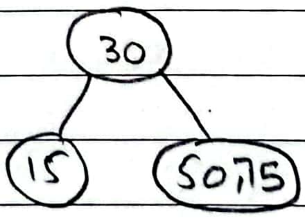


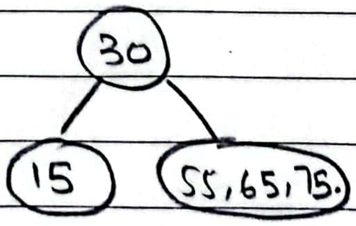




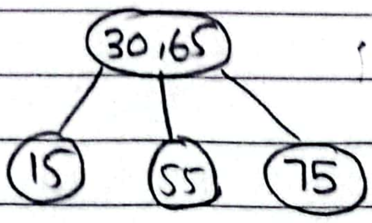
Split because not a 2 node or 3 node.
Make middle parent, less value left child.
and greater value than middle its right child.



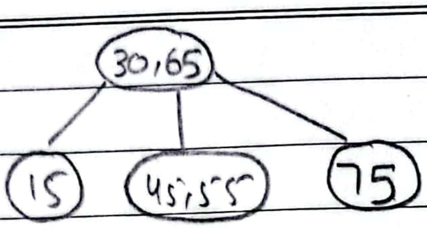


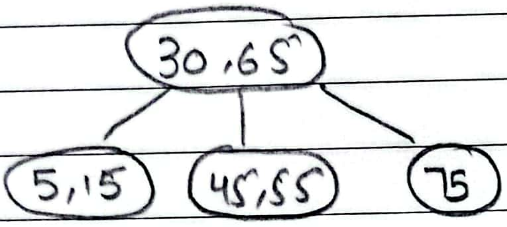


Split 55, 65, 75.

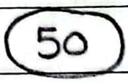


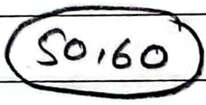
Date:

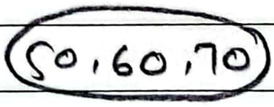




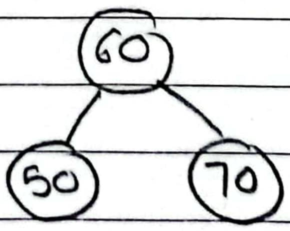
Example # 02: Insert 50, 60, 70, 40, 30, 20, 10, 80, 90, 100

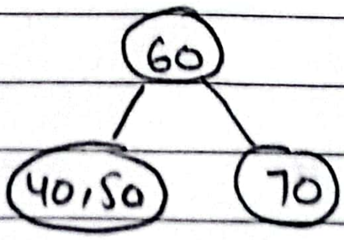


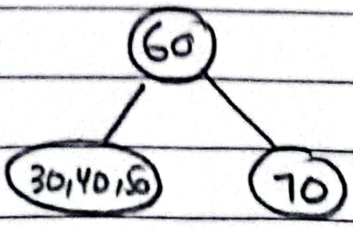




Split because not a 2 node or 3 node.

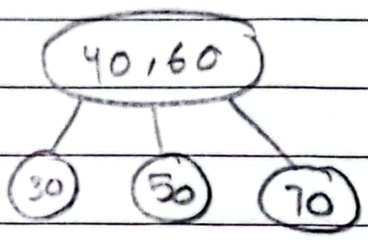


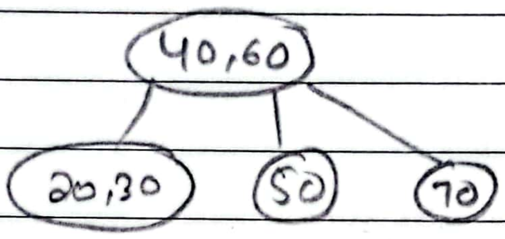


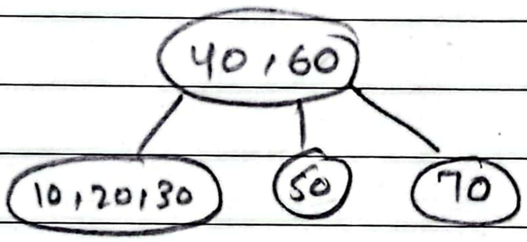


Date:

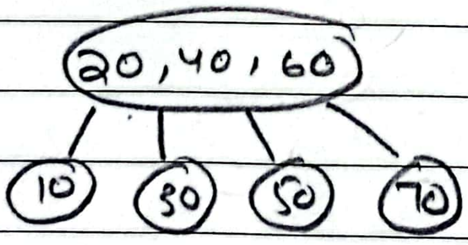
Split 30, 40, 50



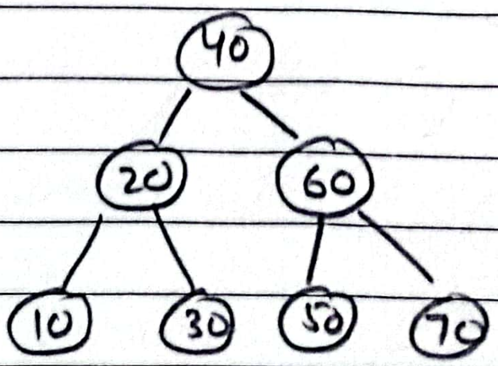




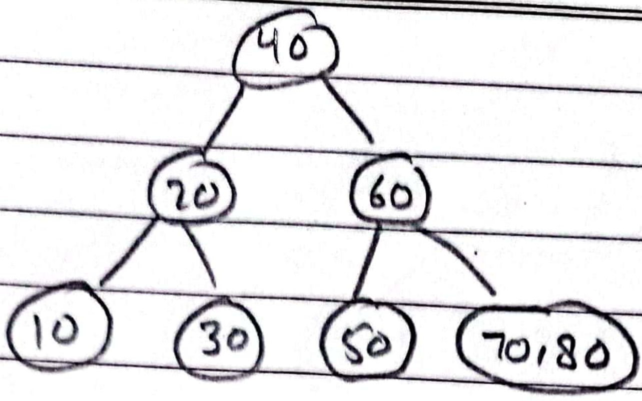
Split 10, 20, 30.

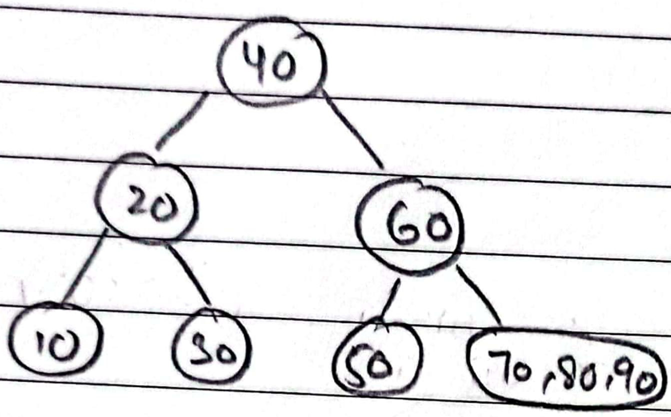


Split 20, 40, 60

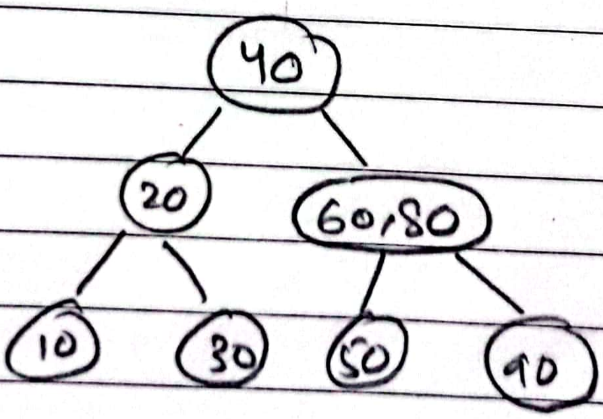


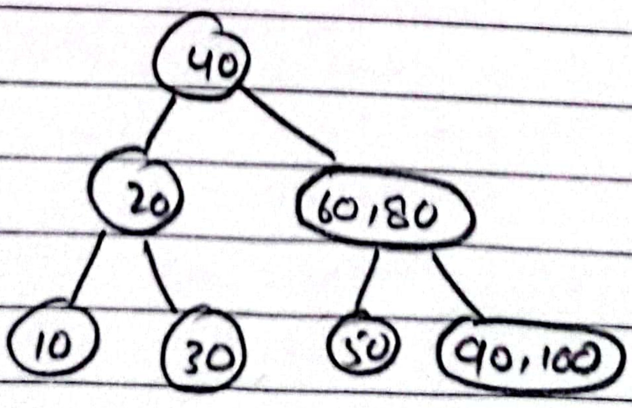
Date:





Split 70,80,90.

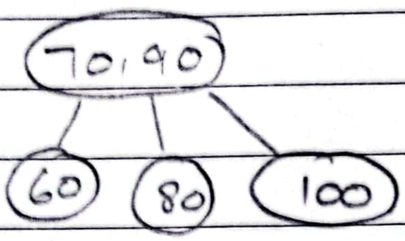




Delete in 2-3 Tree Date:

Example 01:

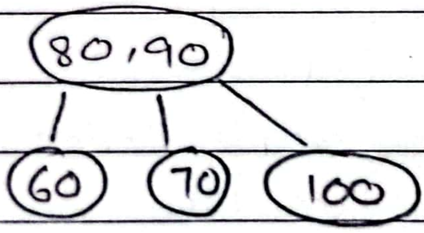
Case I : Delete from root



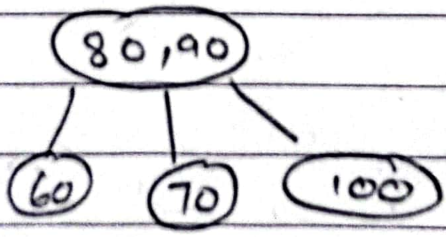
Delete 70?

Replace 70 with its inorder successor.

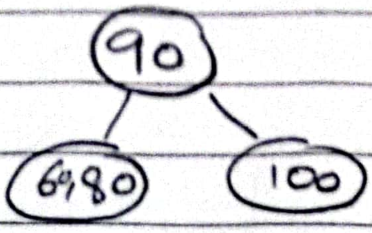
80 is inorder successor, Replace with 70.



Delete 70.

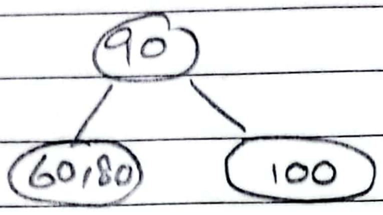


Not a 2-3 tree as not 2 node or 3 node.
Merge 80 and 60.



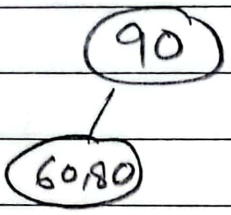
Date:

Case II : Delete leaf

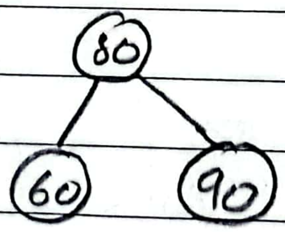


Delete 100?

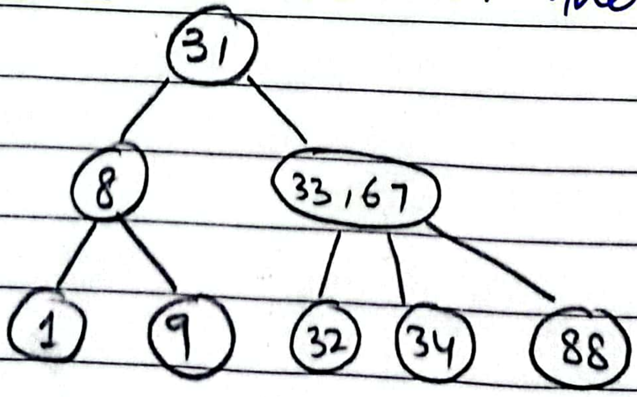
Deleting 100, we get:



Not a 2node or 3 node.
Redistributing in such a way that it becomes a 2-3 tree.



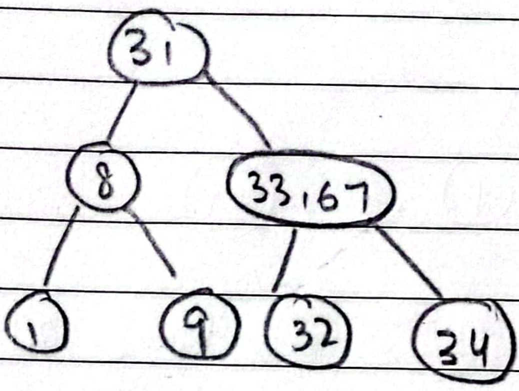
Example 02: Delete 88, then 33, then 31.



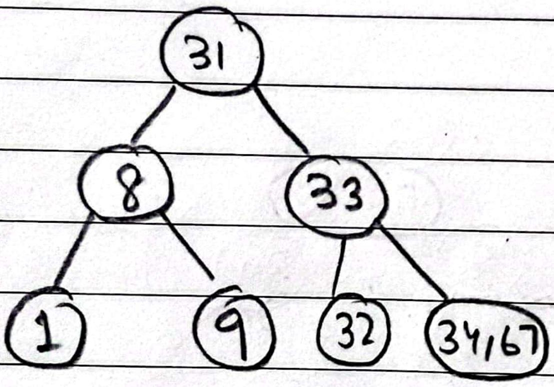
Delete 88, then 33, then 31.

Date:

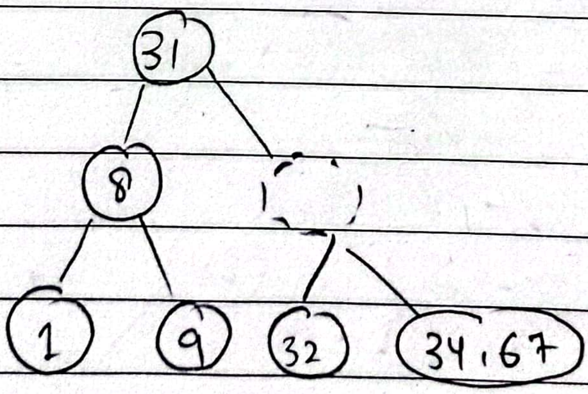
Deleting 88, we get:



merge 34 and 67.



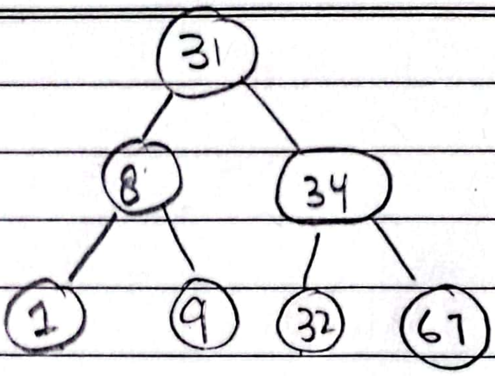
Deleting 33, we get



Redistributing child nodes of 33.

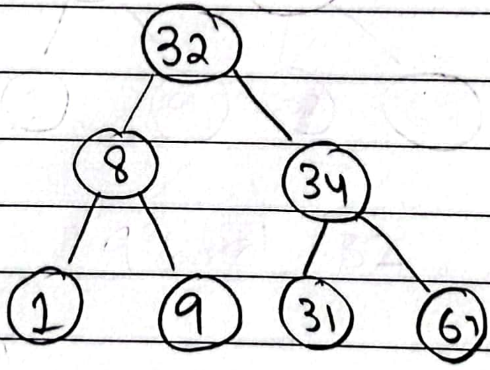
Delete

Date:

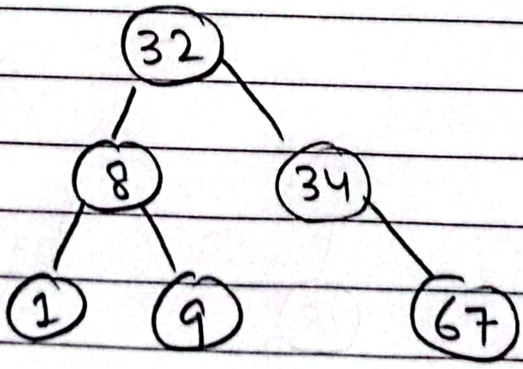


Deleting 31, replace 31 with inorder successor.

Replace 32 with 31.



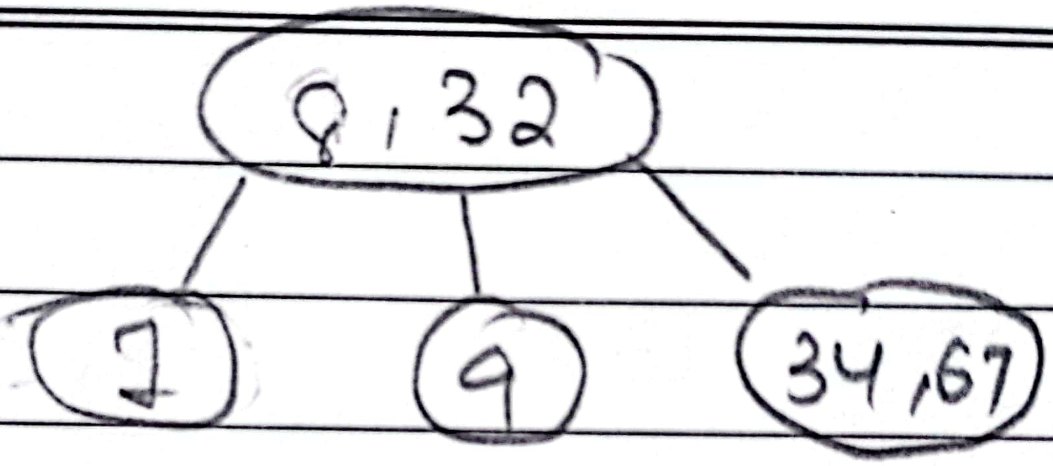
Delete 31.



Not a 2 node or 3 node.
merge 8 and 32, 34 and 67.

e data set.

In BST.



Date:

Search In 2-3 Tree :

Algorithm :-

① Search starts at rood node.

② If root node is 2 node search it as BST :

a) If K is equal to root, return root.
b) If K < root, go to left.
c) If K > root, go to right.

③ If root is a 3 node.

a) If root is equal to any of the 2 keys (Key 1 and Key2), return root.
b) If root < Key 1, go to left.
c) If root > Key 1 and ~~[illegible]~~ root < Key 2 go to middle.
d) If root > Key2, go to right.