Binary Search Tree

* A Binary Search Tree is a binary Tree that may be empty. A nonempty binary search tree satisfies the following properties.

1. Every element has a key (or value) & no two elements have the same key. ∴ all keys are distinct.
2. The keys in the left subtree of the root are smaller than the key in the root.
3. The keys in the right subtree of the root are larger than the key in the root.
4. The left & right subtrees of the root are also binary search trees.

* The number inside a node is the element key.

* The tree shown in the following figure(a) is not a binary search tree even though it satisfies properties 1, 2 & 3. The right subtrees fails to satisfy property 4.

* This subtree in fig (a) is not a binary search tree, as its right subtree has a key value (22) that is smaller than the key value in the right subtree's root (25)

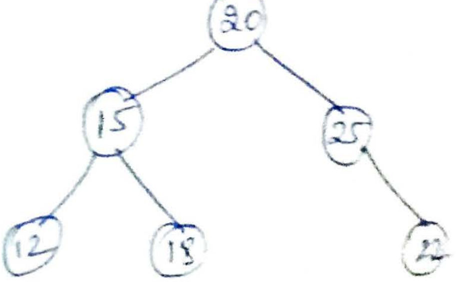


fig (a)

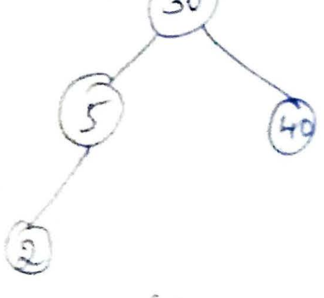


fig (b)

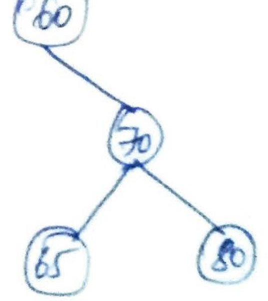


fig (c)

* the binary trees of fig (b) & (c) are bin
search trees.

Binary Search ~~Opo~~ Tree operations

1 Searching :- To search for a ~~k~~ pair with the given key '~~thekey~~'.

Step 1 :- If the root is NULL the search tree contains no pairs & the search is successfull.

Step 2 : Or Compare key with the key ~~of~~ in the root, If the key value is less than the key in the root then no pair in the right subtree can have the key & only left subtree has to be searched.

Step 3 :- If the key is larger then the key in the root, only the right subtree needs to be searched.

Step 4 : If the key equals the key in the root then the search terminates successfully.

2) Inserting an Element

Step 1 : To insert a new pair in-to a binary tree, first determine whether it's key is different from those of existing pairs by performing a search for the key.

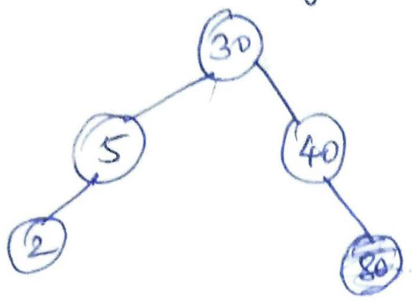
step 2 : If the search is successfull, replace the old value associated with the first pair with the ~~value~~ the value of the second.

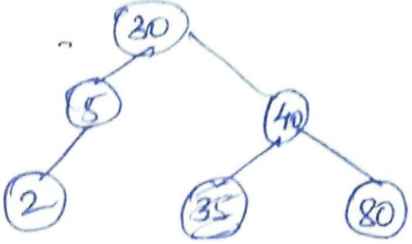
Step 3 :- If the search is unsuccessfull, then the new pair is inserted as a child of the last

5 node examined during the search.

~~2> Deleting an Element.~~

Ex:- Inserting 80 into the tree.



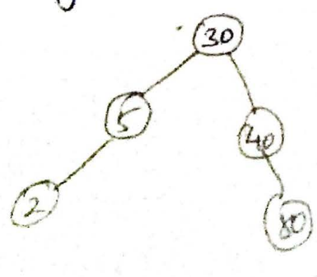


3> Deleting an Element.

* The 3 posibelities for the node p that contains the pair that is to be removed.
(i) p is a leaf (ii) p has exactly one non empty subtree (iii) p has exactly two non empty subtrees

Case i : It is handled by discarding the leaf node & if the discarded leaf was also the tree root the root is set to NULL.

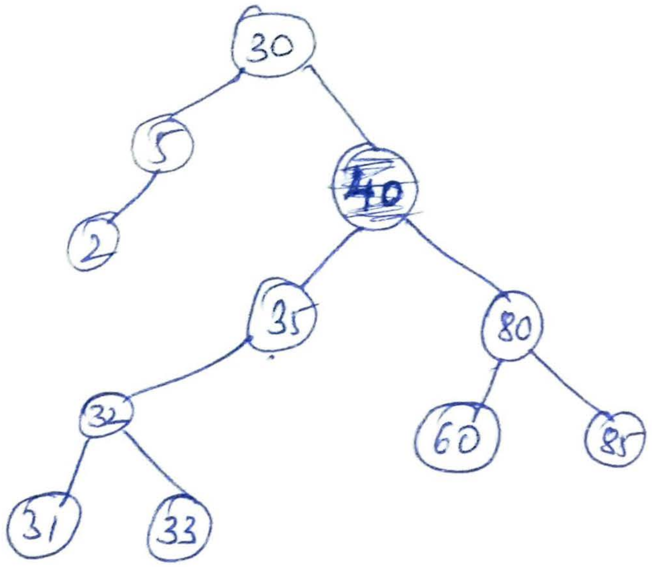
Case ii :- If p has no parent (it is the root), the root of its single subtree becomes the new search tree root. If p has parent then change the pointer from parent so that it points to p's only child.



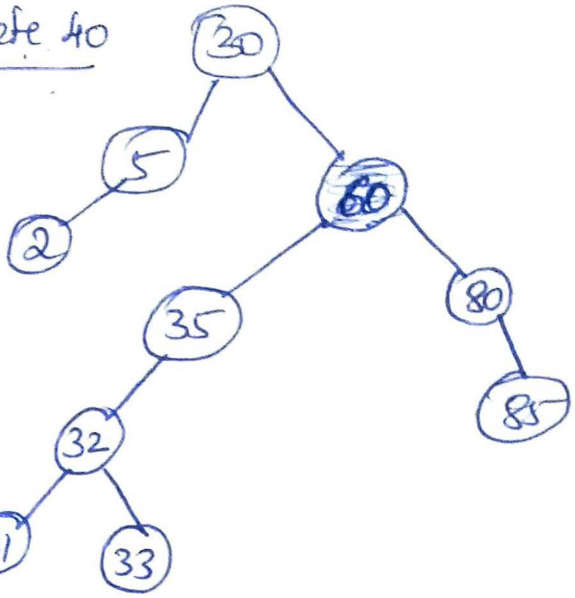
To delete 5, make root 30 to points to the 5's only child 2

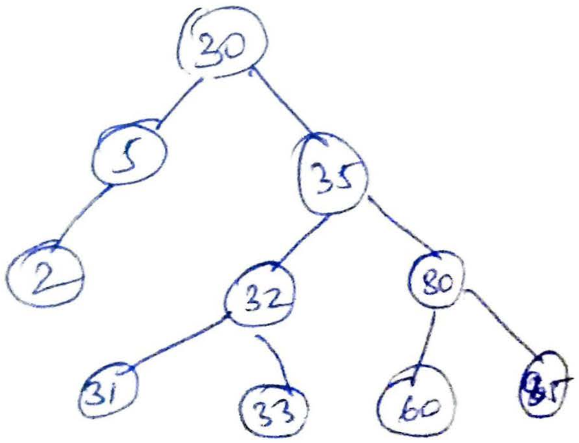
Case iii: To remove a pair in a node that has two non empty subtrees, replace this pair with either the largest pair in its left subtree or the smallest pair in its right subtree.

* The replacing pair is removed from its original node.

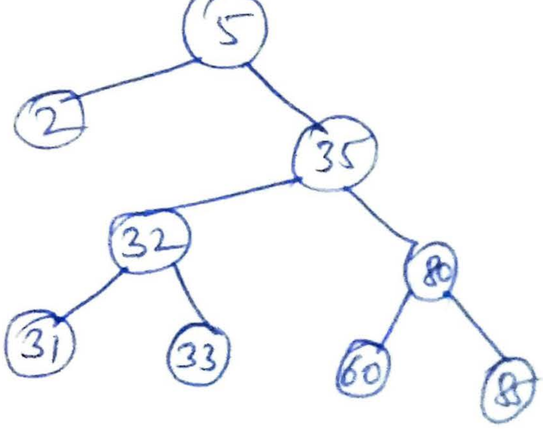


To delete 40





To delete 30


5
2
35
32
80
31
33
60
85