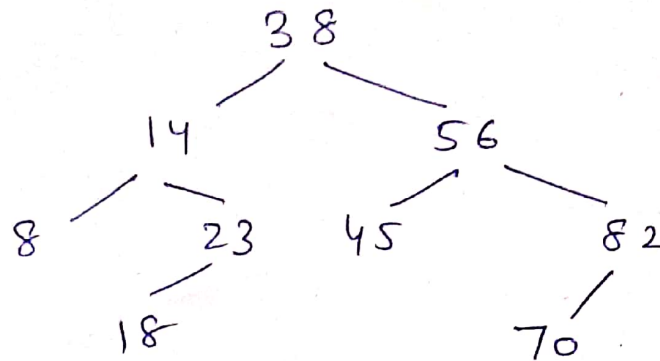


BINARY SEARCH TREE (BST) :- Suppose T is a binary tree.

Then T is called BST if each node N of T has following property :-

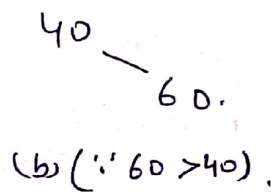
- * The value of N is greater than every value in the left subtree of N and is less than every value in the right subtree of N.
- For e.g.



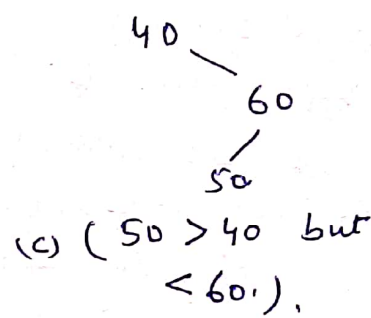
14 < 38 & 14 is left of 38
 56 > 38 & 56 is right of 38
 8 < 14 & 8 is left of 14
 23 > 14 & 23 is right of 14
 70 < 82 & 70 is left of 82

* Construct BST from the six no. given below
 40, 60, 50, 33, 55, 11

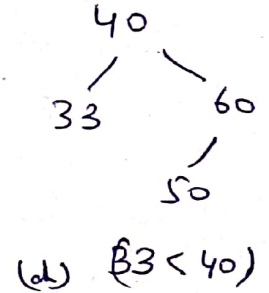
40
 (a) Item = 40.



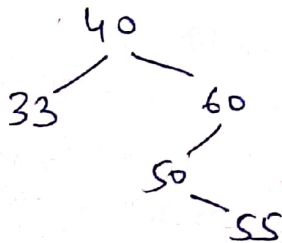
(b) (∵ 60 > 40).



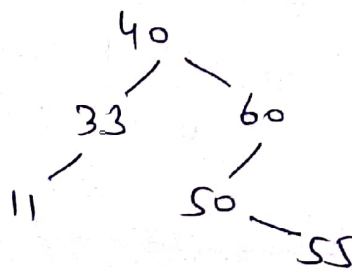
(c) (50 > 40 but < 60).



(d) (33 < 40)



(e) (∵ 55 > 50).



(f) (∵ 11 < 33).

Insertion or

Searching in BST. Suppose we want to search ITEM in a binary search tree T, or inserts ITEM as a new node in its appropriate place in the tree.

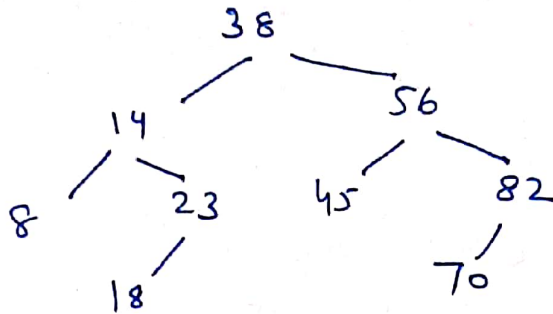
- g) Compare ITEM with root node N of the tree
 - (i) If ITEM < N, proceed to left child of N
 - (ii) If ITEM > N, proceed to right child of N

(b) Repeat Step (a) until one of following occurs

- (i) we meet a node N such that $ITEM = N$, In this case the search is successful.
- (ii) we meet an empty subtree, which indicates that the search is unsuccessful, and we insert $ITEM$ in place of the empty subtree.

In other words we can say that :-
 Root R की खोज करने के लिए Tree में Traverse करें
 यदि कोई Node ITEM के बराबर मिले तो
 उस Node पर Insert करें।

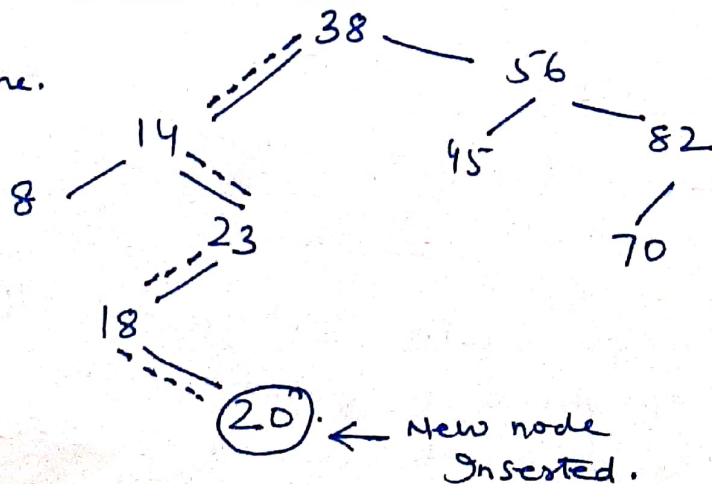
For e.g Consider tree.



Suppose Item = 20.

1. Compare 20 with root 38. As $20 < 38$ we will move to left child of 38 i.e 14
2. Compare 20 with 14. As $20 > 14$ we will move to Right child of 14 which is 23.
3. Compare 20 with 23. As $20 < 23$ we will move to Left child of 23 which is 18
4. Now Compare 20 with 18. Since $20 > 18$ we move to Right child of 18. But 18 does not have a right child, Insert 20 as right child of 18.

ie processing is shown in Dotted line.



Ques. Make Binary Search Tree from the following nodes.
14, 10, 17, 12, 10, 11, 20, 12, 18, 25, 20, 8, 22, 11, 23