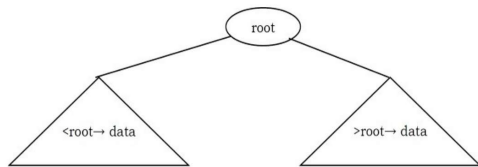
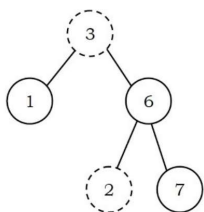
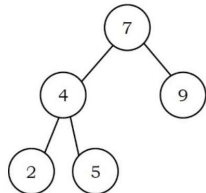


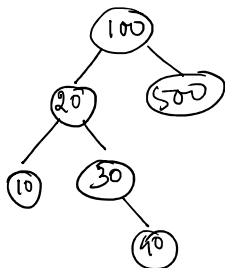
In binary search trees, all the left subtree elements should be less than root data and all the right subtree elements should be greater than root data. This is called binary search tree property. Note that, this property should be satisfied at every node in the tree.



Which one is BST????



Insert 100, 20, 30, 500, 10, 40 in BST.



Rule for inserting new node in BST:

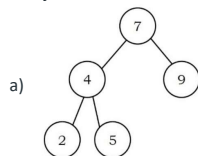
1. Start from root node.
2. If node.value < root.value, move in left direction.
3. If node.value > root.value, move in right direction.

B-Tree (Balanced tree)

In a B-Tree of order m, each node can have up to m children or m pointers and m-1 keys (values).

Binary search tree:

1. Order is 2. Why? Because each node can have maximum 2 children or 2 pointers and each node has only 1 value or 1 key.

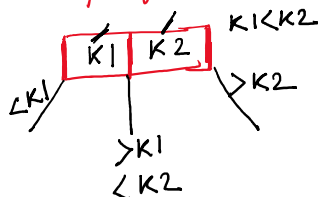


K1 → Boundary represent pointers.

Max
Order = 2 → No. of children = 2.

No. of keys in a node = 1

Order = 3, Max-no of children = 3
Max No. of keys in a node = 3-1 = 2

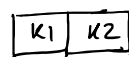


Insertion in B-tree

1. Insertion of new nodes should be always at last level.
2. B tree grows in upward direction.
3. Keys should be sorted inside a node.

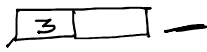
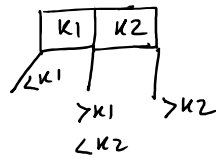
Insert keys in B-Tree of order 3.

3 2 1 5 4 6 7 8 9 10



3 2 1 5 4 6 7 8 9 10

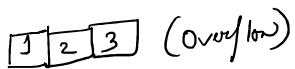
order, $m=3$, Max no. of child/pln $=3$
No of keys in a node $=3-1=2$



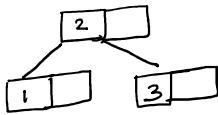
⇓



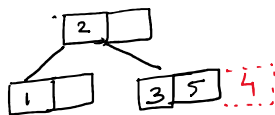
⇓



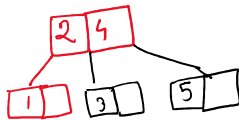
⇓



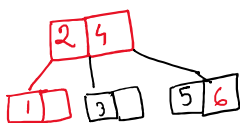
⇓ 5, 4



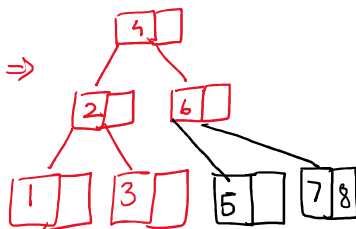
⇓ 4



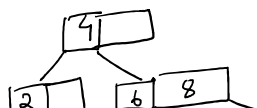
⇓ 6



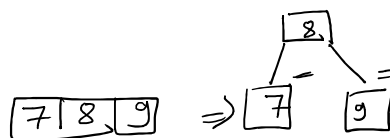
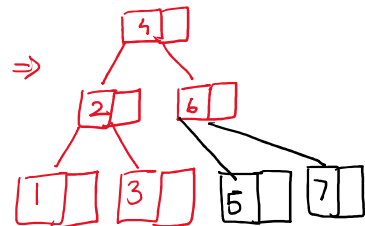
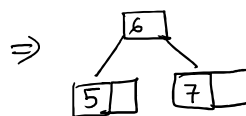
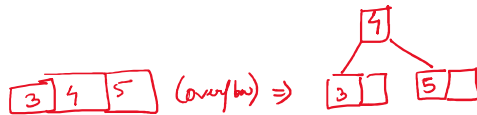
⇓ 7.

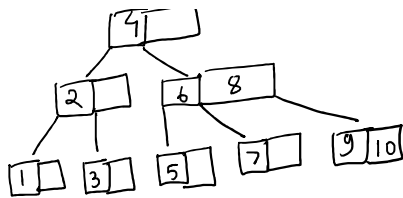


⇓ 9

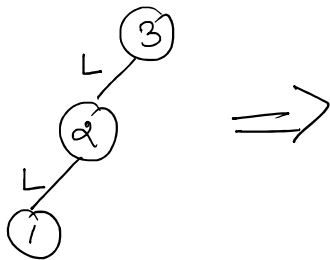


- ① Find middle
- ② Split on the basis of middle element.





Rotation-



Minimum = left
 Maximum = Right
 Remaining = Root.

