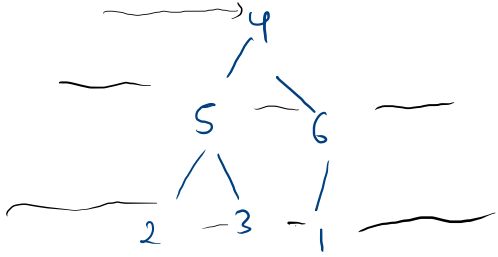
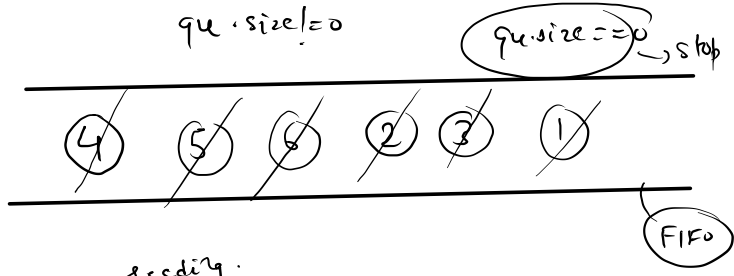
Revision

4 5 6 2 3 1

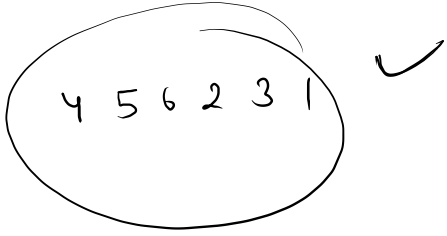
Level Order.





seeding.

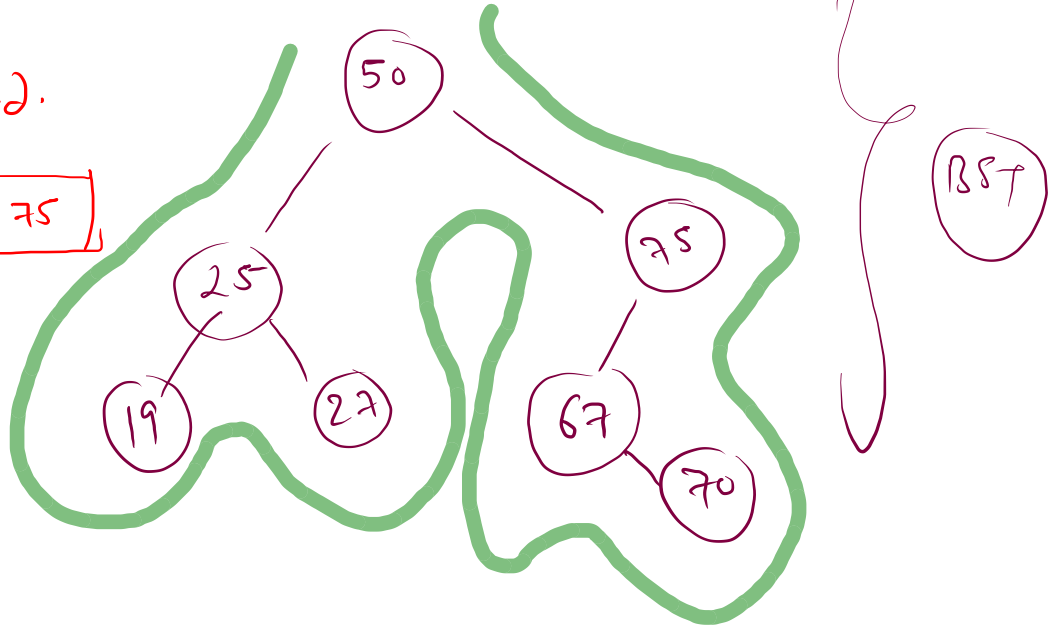
algo → BFS



Revision. BST → Binary Search Tree.

inorder of BST : sorted.

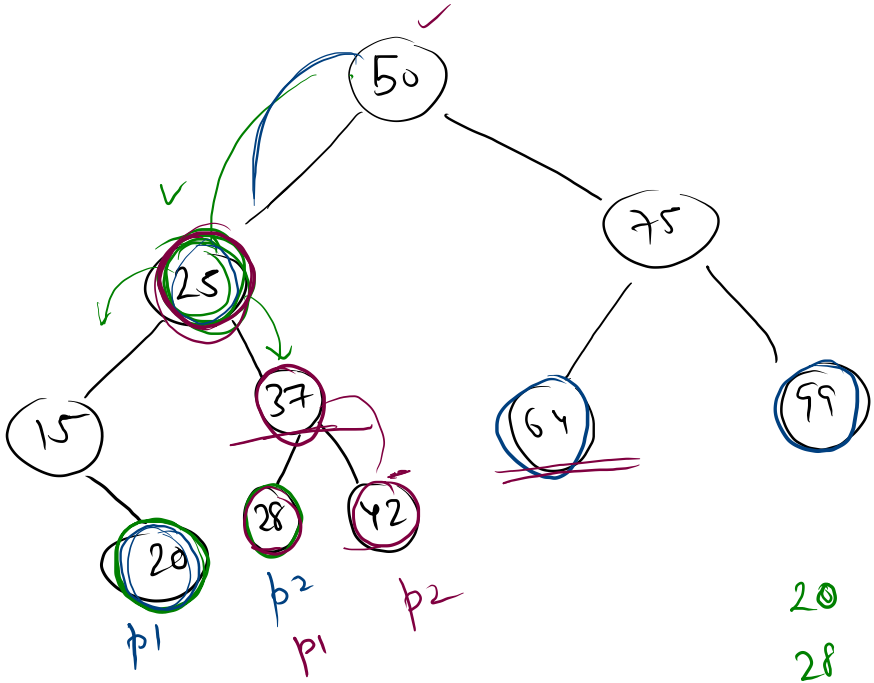
19 25 27 50 67 70 75



LCA) Lowest Common Ancestor.

(BT / BST).

BST.



64, 99 → 75

37, 64 → 50

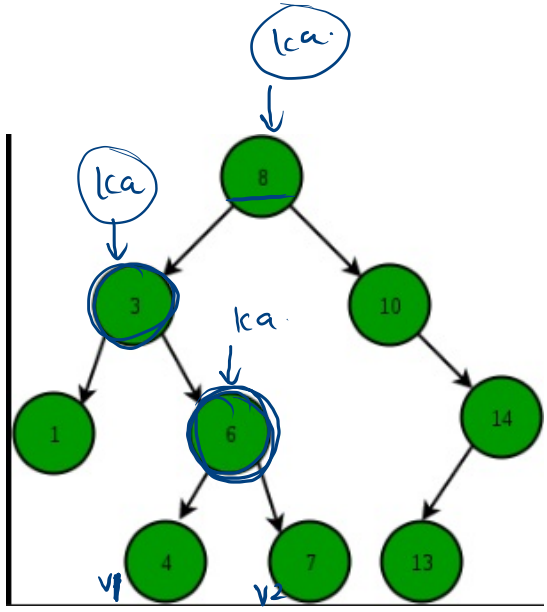
28, 42 → 37

20 15 25 50

28 37 25 50

BST. : LCA

LCA =



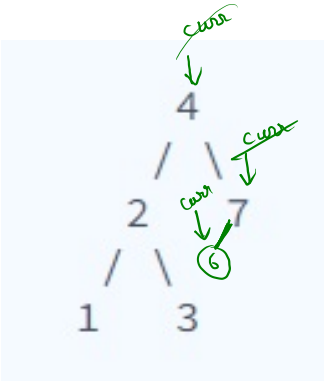
v1 > lca
v2 > lca .

v1 < 8 ⇒ 4 < 8
v2 < 8 ⇒ 7 < 8

v1 > lca } right
v2 > lca

v1 < lca } left
v2 < lca

Binary Search Tree : Insertion

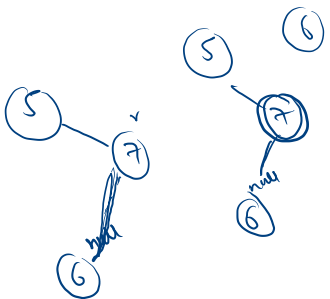


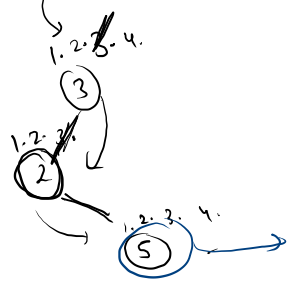
curr == node

↳ create node

7.left = 6

```
public static Node insert(Node root,int data) {
    if(root == null){
        return new Node(data);
    }
    if(root.data > data){
        // left
        root.left = insert(root.left, data);
    }
    else if(root.data < data){
        //right
        root.right = insert(root.right, data);
    }
    return root;
}
```





isBST?

```java
int prev = Integer.MIN_VALUE;

boolean checkBST(Node root) {
    if(root == null){
        return true;
    }

    boolean la = checkBST(root.left);
    if(la == false)
        return false;
    //process

    if(prev >= root.data){
        return false;
    }

    prev = root.data;

    boolean ra = checkBST(root.right);
    if(ra == false)
        return false;

    return true;


}
```