**root=20, root.left = deleteNode(15, 17)**

**root=15, root.right = deleteNode(18, 17)**

**return = null , 17==17**

**root=18, root.left = deleteNode(16, 17)**

**deleteNode(20,17)**

**root=16, root.right = deleteNode(17, 17)**

**root.right=null**

**root.left =16**

**root.right=18**

**root.left =15**

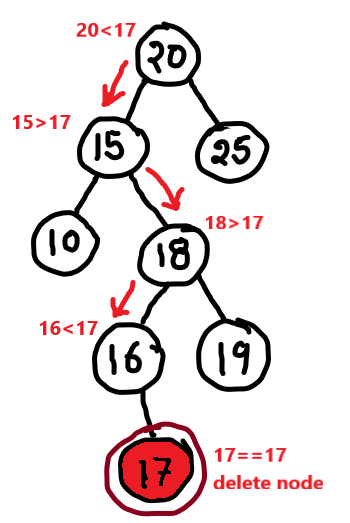
**if (root.left == null && root.right == null)**

**{**

**root = null;**

**return root;**

**}**



**root=20, root.left = deleteNode(15, 18)**

**root=15, root.right = deleteNode(18, 18)**

**return = 16 , 18==18**

**deleteNode(20,18)**

**root.right=16**

**root.right=18**

**root.left =15**



**if (root.right==null)**

**{**

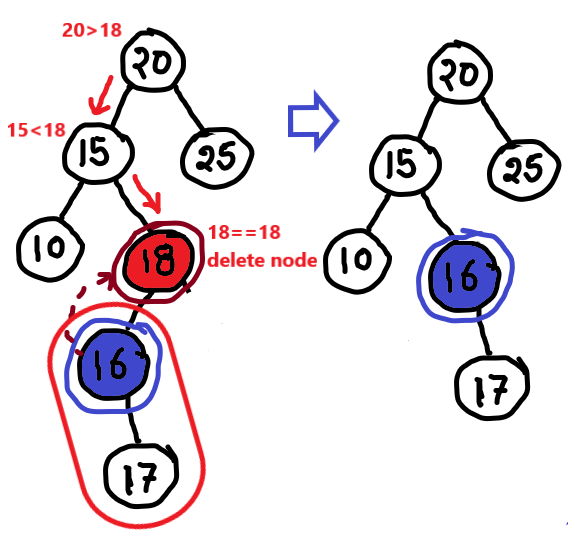
**Node temp = root.left;**

**root.left = null;**

**root = temp;**

**return root;**

**}**





**root=30**

**Node minimum = findMinimum(root.right);**

**root.data = minimum.data;**

**deleteNode(root.right, minimum);**

**deleteNode(30,30)**

**root.data=50 replaced**

**deleteNode(90,50)**

**end after iteration done**

**root.right=90**

**root.right=50**

**root.right=55**

**50**

**90**

**private Node findMinimum(Node node) {**

**if (node.left == null)**

**return node;**

**else**

**return findMinimum(node.left);**

**}**

**root=90, root.right = deleteNode(90, 50)**

**return = 55 , 50==50**

**root=50, root.right = deleteNode(50, 50)**

