

Binary Search Tree II

RM.Periakaruppan
Dept of AM&CS
PSG Tech

Binary Search Tree – Insert Algorithm

```
/* Insert Tree: insert a new element with key e */
/* in the tree.*/
p = tree;
q = null
while (p! = null)
{
  If (e = key (p))
    return (p);
  q = p
  If (e<key (p))
    p = left (p);
  else p = right (p);
}
```

Binary Search Tree – Insert Algorithm (Contd)

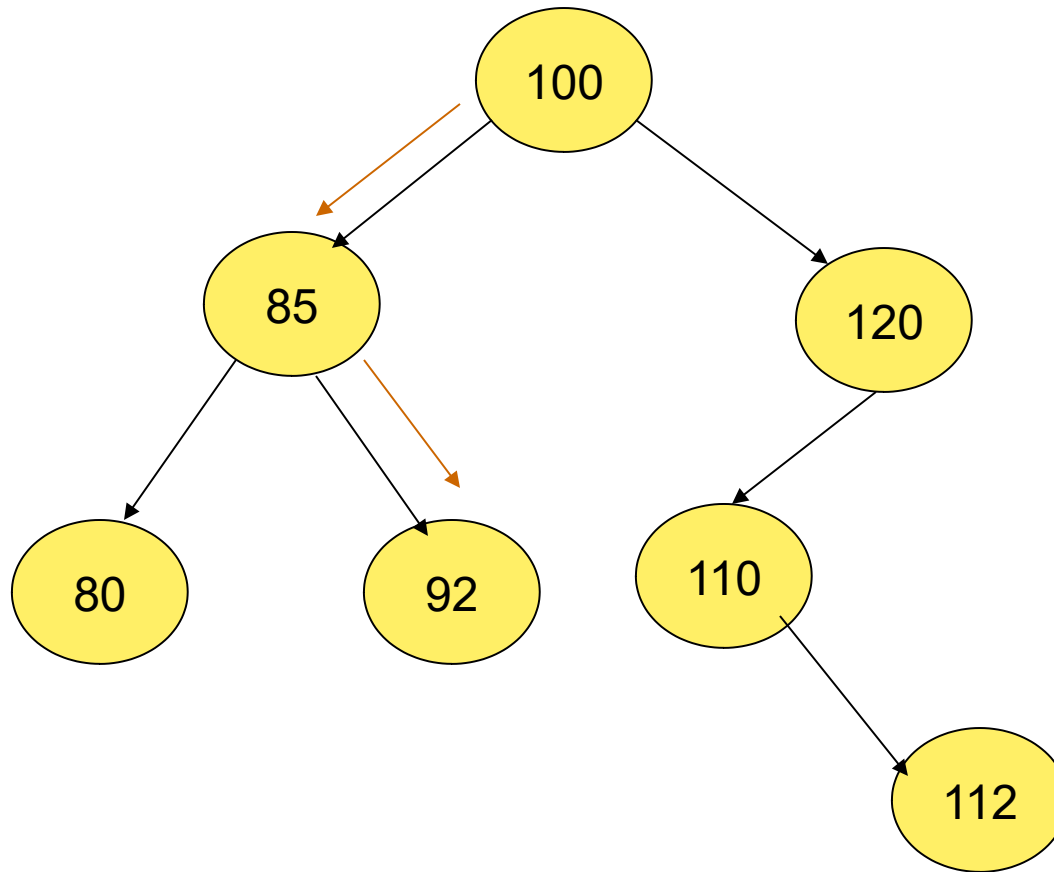
```
create memory for newnode
key(newnode) = e;
left(newnode)=null;
right(newnode)=null;
If (q = null)
root = newnode;
else If (e<key (q))
left (q) = newnode;
else right (q)= newnode;
```

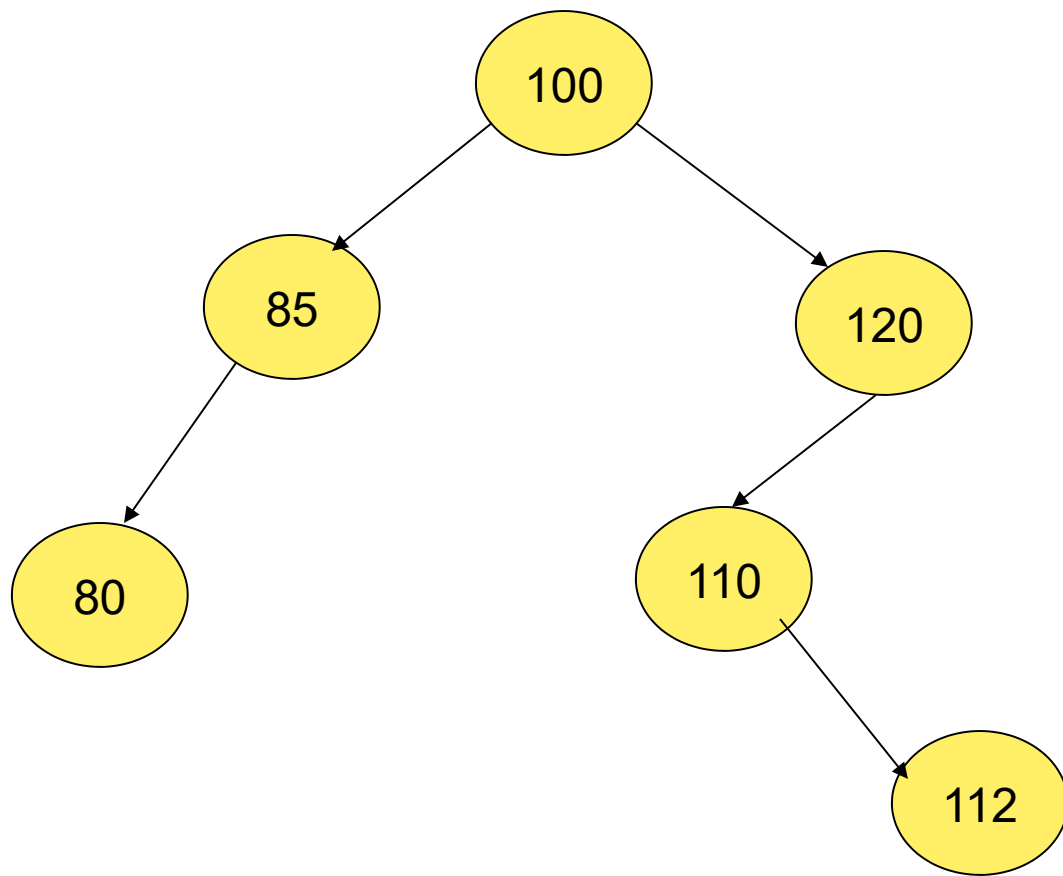
Deleting an Element

- **Deleting a leaf:** Deleting a node with no subtree is easy, as we can simply remove it from the tree.
- **Deleting a node with one subtree:** Delete it and replace it with its child.
- **Deleting a node with two subtrees:** Suppose the node to be deleted is called N . We replace the value of N with either its in-order successor (the left-most child of the right subtree) or the in-order predecessor (the right-most child of the left subtree).

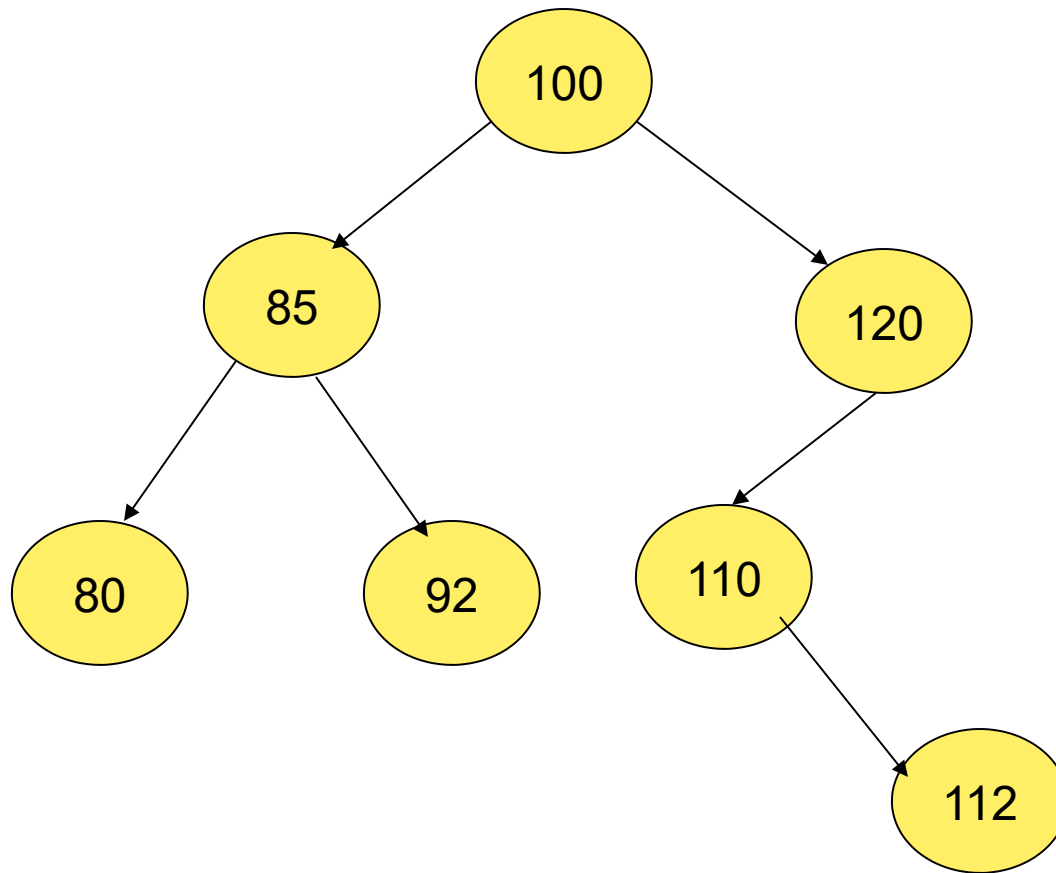
DELETION

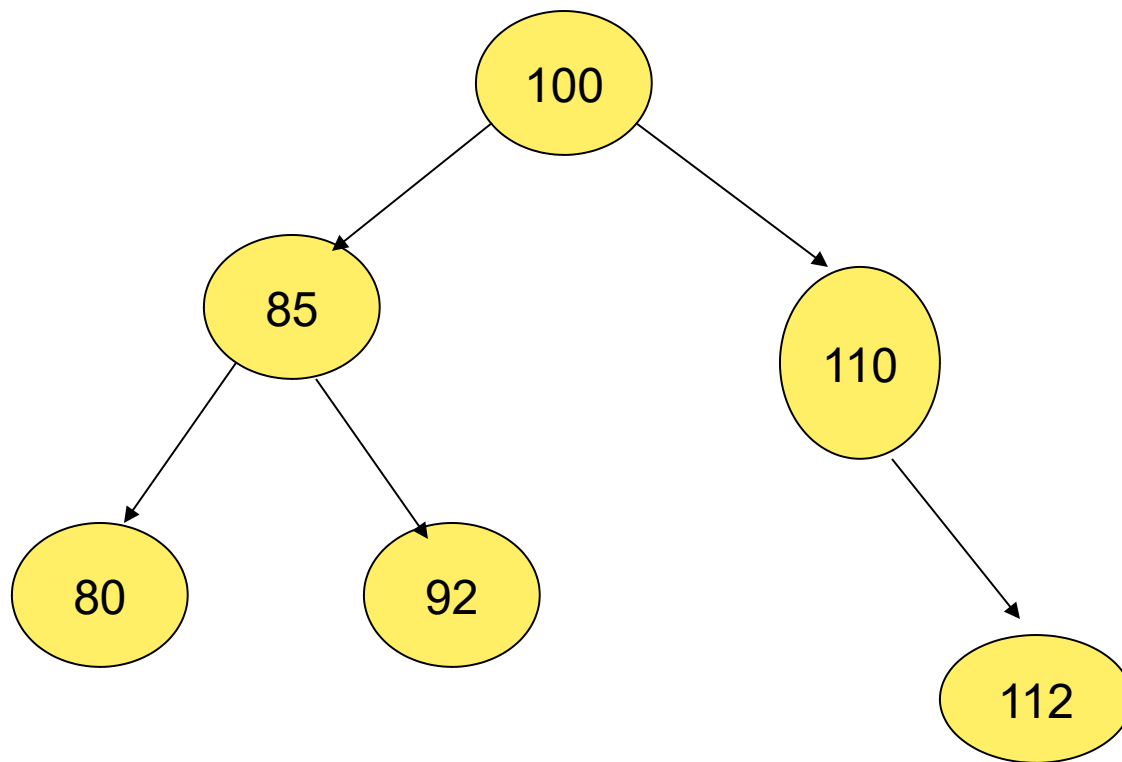
P is a leaf node DELETE 92





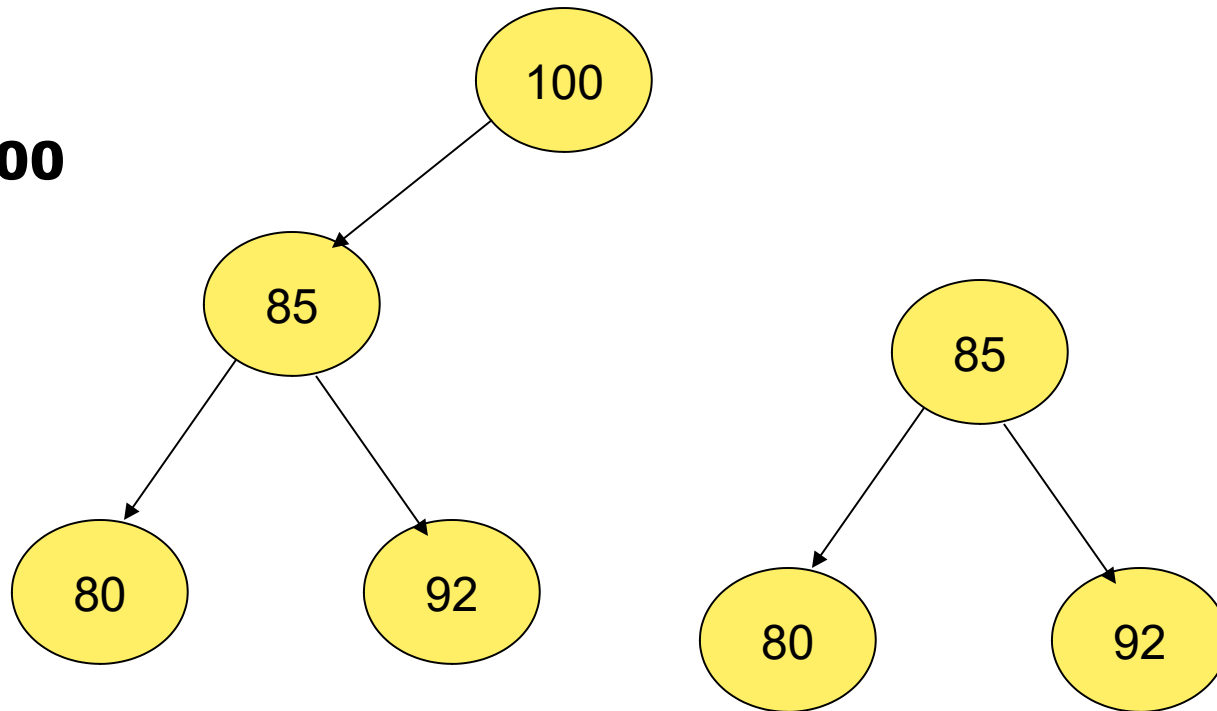
p has exactly one subtree Delete 120





Node to be deleted (p) is a root node which has exactly one subtree

Delete 100



P has exactly two subtrees Replace P by the largest element in the left subtree or the smallest element in the right subtree

DELETE 120

