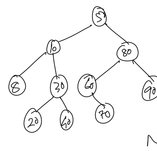
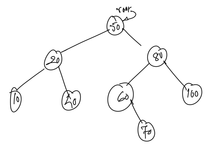


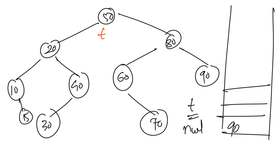
pre 50 10 5 30 20 40 80 60 70 90
In 5 10 20 30 40 50 60 70 80 90
pos 5 20 40 30 10 70 60 90 80 50

Node
int data
Node left, right

BST
Node root



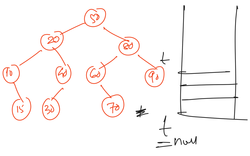
```
begin loop   t = r;
             if d < t.gd()
                 if t.gL() == null r
                     t.SL(nn)
                     r ← true
                 }
                 t = t.gL()
             else
                 if t.gR() == null r
                     t.SR(nn)  r ← true
                 }
                 t = t.gR()
end loop
```



```
t = root
w (t != null || s.isEmpty())
    w(t != null)
        sop(t)
        push(t)
        t = t.gL()
    t = pop()
    t = t.gR()
```

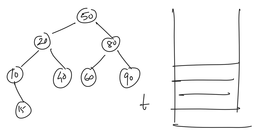
50 20 10 15 40 30 80 60 70 90

w((t == null && s isEmpty))



```
t = r
w (t != n || s.isEmpty())
    w(t != null)
        push(t)
        t = t.gL()
    t = pop()
    Sop(t)
    t = t.gR()
```

10 15 20 30 40 50 60 70 80 90

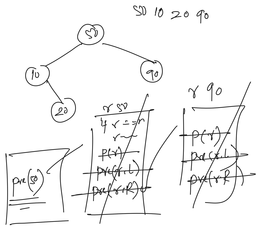


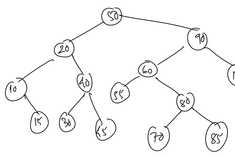
```
w(
    w(t != null)
        push(t, L)
        t = t.gL()
    pair = pop()      50 R
    if pair.flag == L
        t = pair.node.right
        pair.flag = R
        push(pair)
    else
        Sop(pair.node)
```

15 10 40 20 60 90 80 50

```
Pair
node
flag 'L' 'R'
```

50 10 20 90





55

① Locate del along with parent

② Check if del is terminal
if del is terminal
↳ delete del
return true

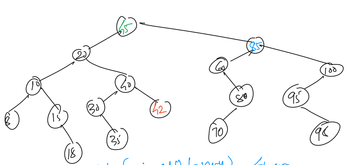
③ if del. is non terminal
→ make del a terminal node &
then delete it
→ Shift del down the tree (Swap)
→ Select node with max value from lef subtree of del
or
Select node with min value from Right sub tree

```
locate node to be deleted
    del = root
    while( del.gd() != data) {
        if( data < del.gd() ) {
            p = del
            del = del.gL()
        }
        else{
            p = del
            del = del.gR()
        }
    }
② if ( del.gR() == null && del.gL() == null ) {
        if ( p.gL() == del ) {
            p.SL(null)
        }
        else
            p.SR(null)
        return true
    }
```

root → null 50

P (50) d

```
if( del == root ) {
    root = null.
    return true
}
```



~~50~~
~~55~~
55

p = del
min = d.gR()

while(min.gL() != null)
p = min
min = min.gL()

Swap
min & d

d = min