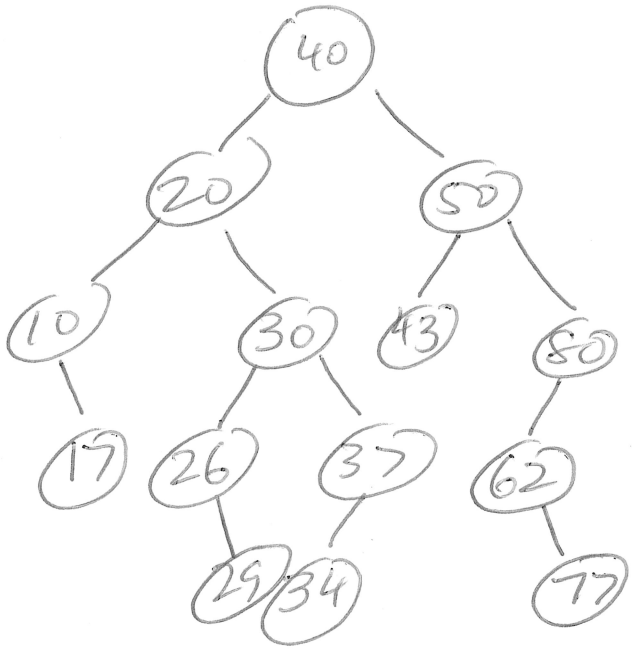
# COP 3502 - 10/25/23

(0) Junior Knights Reminder

(1) Binary Search Trees
- insert
- Tuesday's : Ceiling Function
- delete
- look at posted BST

Insert 29

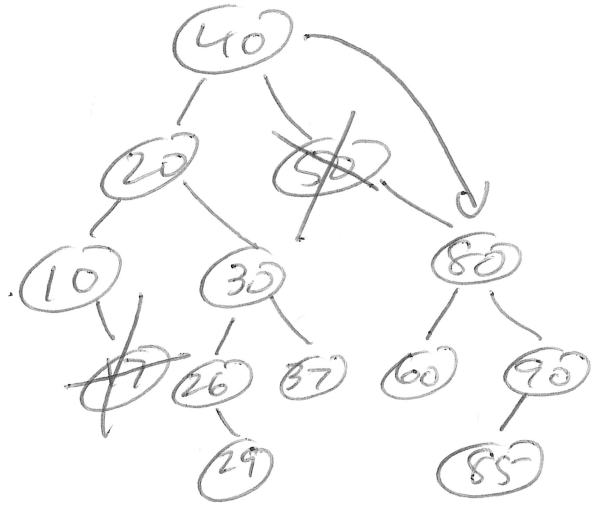



you figure out what's in here!

```
node* insert(node* root, int value) {
    if (root == NULL) return makenode(value);
    if (value < root->data)
        root->left = insert(root->left, value);
    else
        root->right = insert(root->right, value);
    return root;
}
```

O(h) ← height of tree

# Delete




Easy case (leaf node)

Delete 17

→ easy just set
10 ~~left~~ right to NULL +
free node

go down tree.
ID parent of node
to delete + which dir
the deleted node is.
set approp pointer NULL
free node.

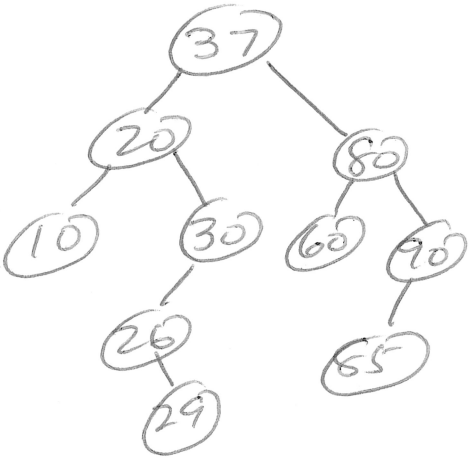
Next case Delete 50 (1 child)
need parent of deleted node
need 1 child deleted node
patch parents L/R ptr to
the L/R child depending
on the case.
(4) cases!

Deleting 40 (2 child)

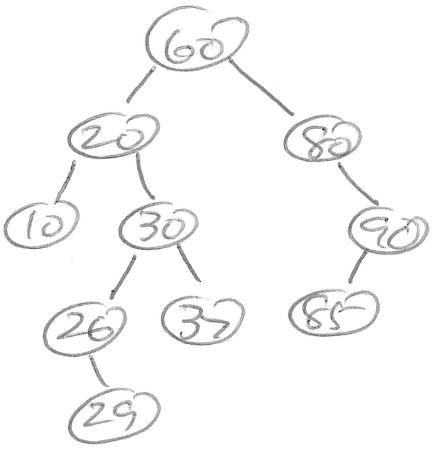
Don't delete physical node
w/ 40!!!
Replace with either
(a) max left
(b) min right

Physically delete OLD
node of (a) or (b)
which is guarenteed not
to have 2 children

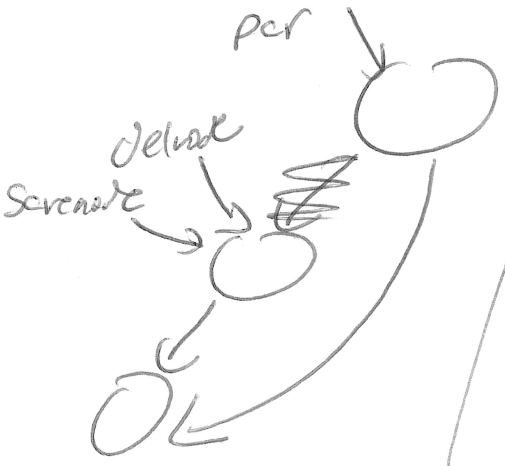
Del 40 via 37




Del 40 via 60




# LL case




```
par->left = par->left->left;
par->left = Serenode->left;
par->left = delnode->left;
```