

COP3502 10/27/21

Attendance = 8091 ~~411306~~

① Study Group turn in

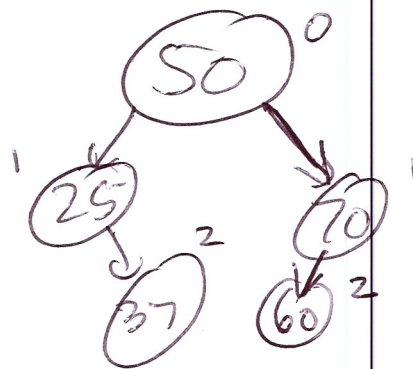
② Binary Trees

Insert

Delete

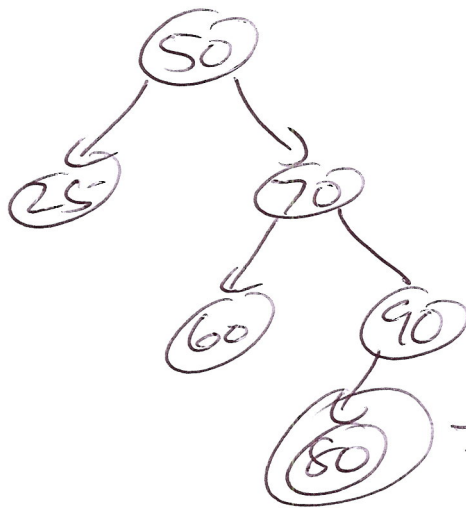
Functions

Insert



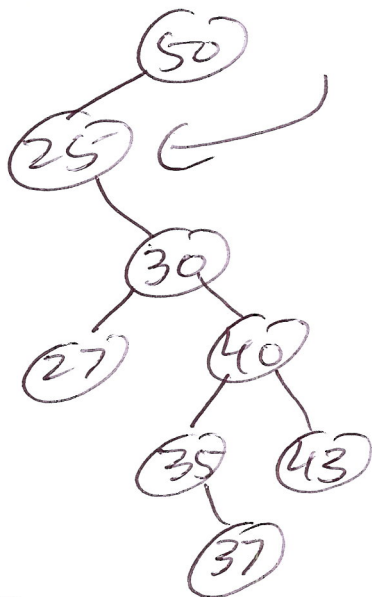
```
binTreeNode* insert(binTreeNode* root, int val) {  
    if (root == NULL)  
        return createNode(val);  
    if (val < root->data) {  
        if (root->left == NULL)  
        root->left = insert(root->left, val);  
    }  
    else  
        root->right = insert(root->right, val);  
    return root;  
}
```

Delete



easy case - leaf node

(1) free node
(2) set 90's left ptr to null.

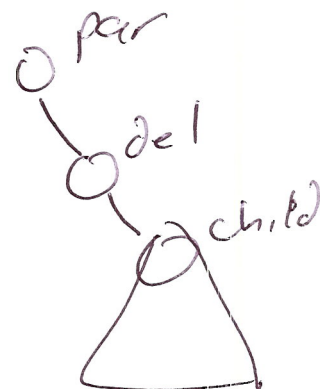
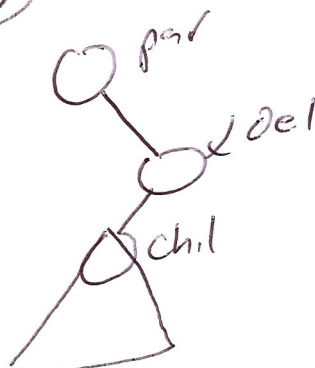
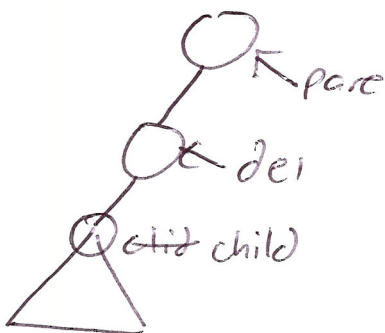


Del 25

one child case

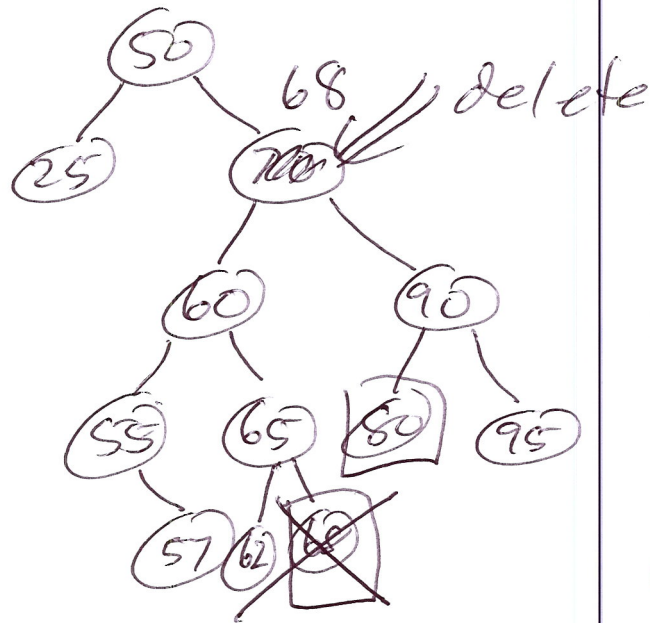
link parent's L or R
to del node's L or R

4 CASES



Order matters - don't lose rest of the tree!

Two child case



Max Val on left has no right child.

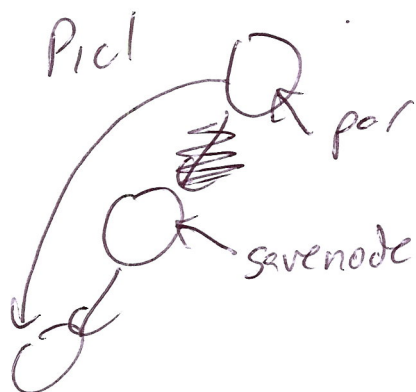
Min Val on right has no left child.

max val - Left
min val - Right

(1) Overwrite value at node to delete with one of these 2

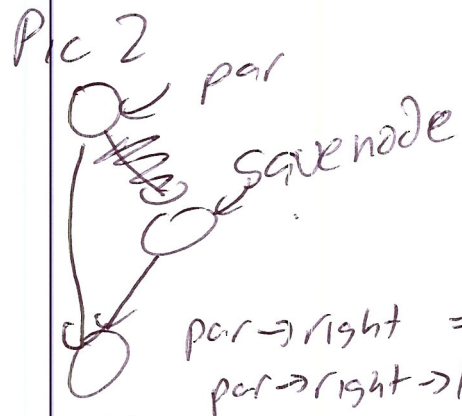
(2) Physically delete the old location of the value copied

One child has Only Left Child



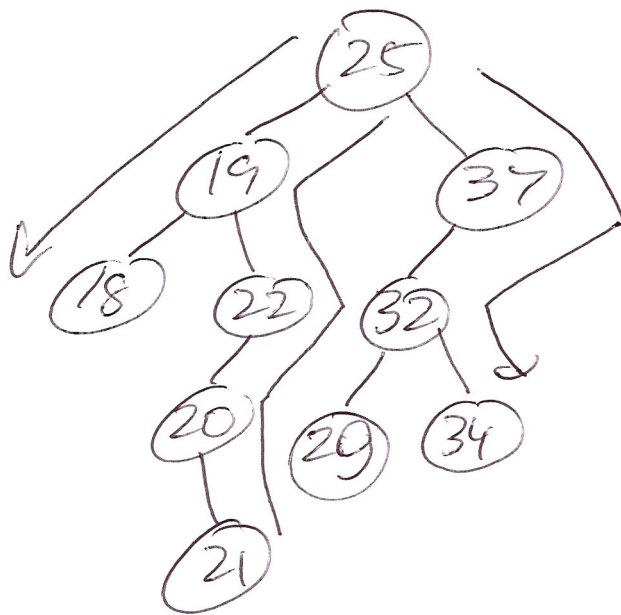
$par \rightarrow left = par \rightarrow left \rightarrow left;$

$free(savenode)$



$par \rightarrow right = par \rightarrow right \rightarrow left;$

Random Problem



$$LL = 25 + 19 + 18 = 62$$

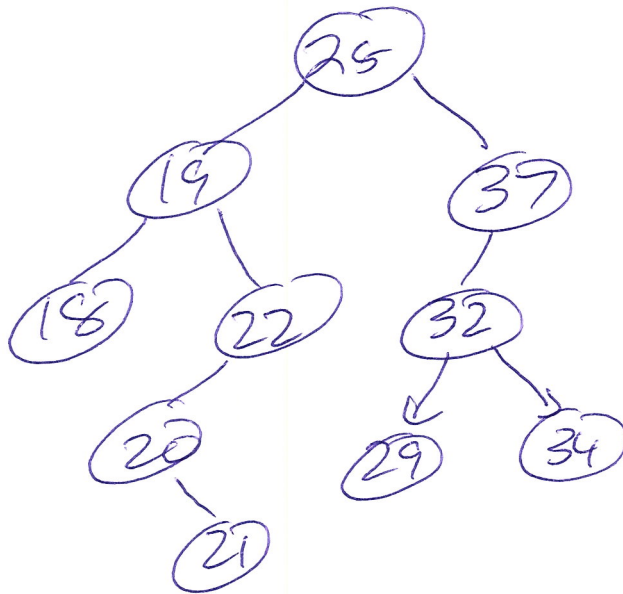
$$LRLR = 25 + 19 + 22 + 20 + 21 = 107$$

$$b. \quad RLR = 25 + 37 + 32 + 34 = 128$$

```
int maxgold(bintree node* root) {  
    if (root == NULL) return 0;  
    int leftScore = maxgold(root->left);  
    int rightScore = maxgold(root->right);  
    if (leftScore > rightScore)  
        return leftScore + root->data;  
    return rightScore + root->data;  
}
```

Afternoon

Problem



Climb with
max elevation
change

$$\begin{aligned} RLL &= |37-25| + \\ &\quad |32-37| + \\ &\quad |29-32| \\ &= 12 + 5 + 3 = 20 \end{aligned}$$

Idea:

recursively go left, right.

build our path by adding

the cost of the hike to left
or right so

$$|25-19| + \text{ansLeft} \quad \text{VS.}$$

$$|25-37| + \text{ansRight}$$

CHOOSE THE LARGER ANSWER