

Even More AVL Trees

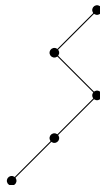
Today's announcements:

- ▶ PA2 out, due March 1, 11:59p

Today's Plan

- ▶ AVL tree code
- ▶ AVL tree remove

A tree is **skinny** if #nodes at each depth is as small as possible.
How many skinny binary trees of height h are there?



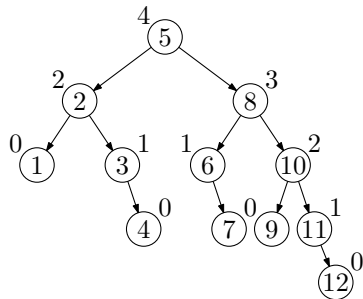
Even More AVL Trees

Today's announcements:

- ▶ PA2 out, due March 1, 11:59p

Today's Plan

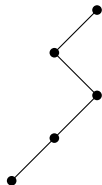
- ▶ AVL tree code
- ▶ AVL tree remove



A tree is **skinny** if #nodes at each depth is as small as possible.

How many skinny binary trees of height h are there?

AVL trees



Insert Algorithm

1. Find location for new key.
2. Add new leaf node with new key.
3. Go up tree from new leaf searching for imbalance.
4. At lowest unbalanced ancestor:

Case LL:  rotateRight

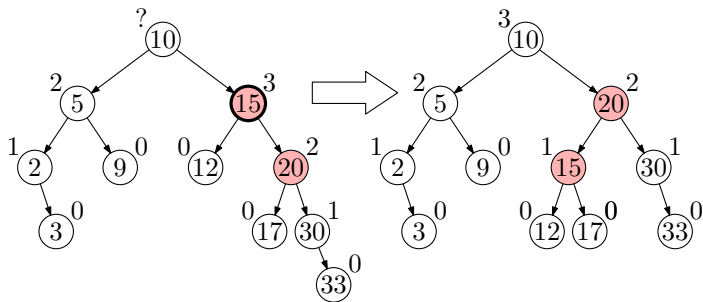
Case RR:  rotateLeft

Case LR:  rotateLeftRight

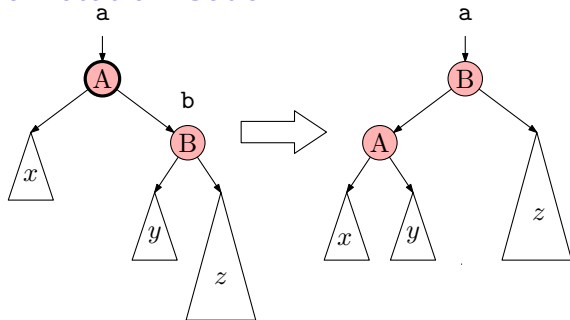
Case RL:  rotateRightLeft

The case names are the first two steps on the path from the unbalanced ancestor to the new leaf.

Case RR: rotateLeft



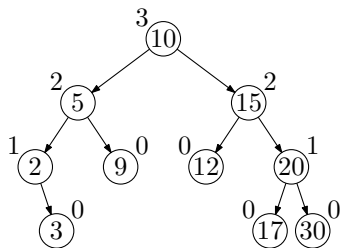
Single Rotation Code



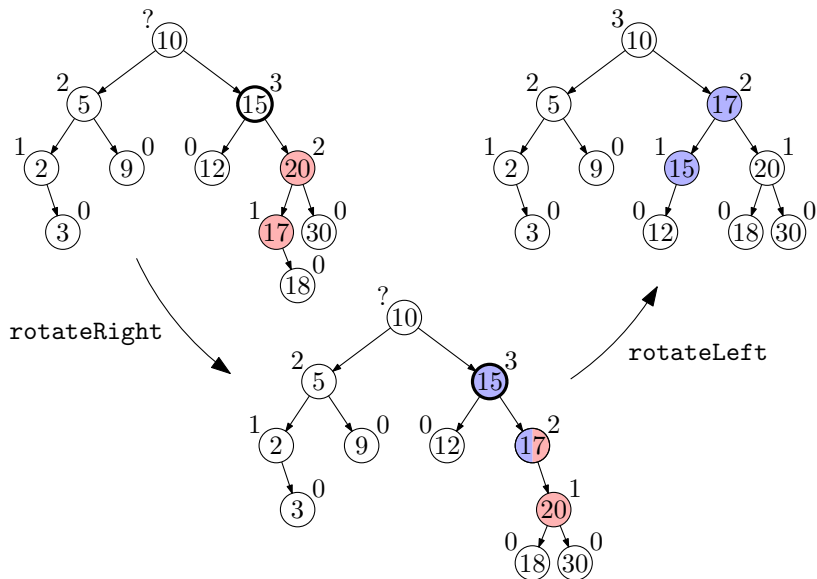
```
void rotateLeft( Node *& a ) {  
    Node * b = a->right;  
    a->right = b->left;  
    b->left = a;  
    updateHeight(a);  
    updateHeight(b);  
    a = b;  
}
```

Insert: Imbalance Case RL

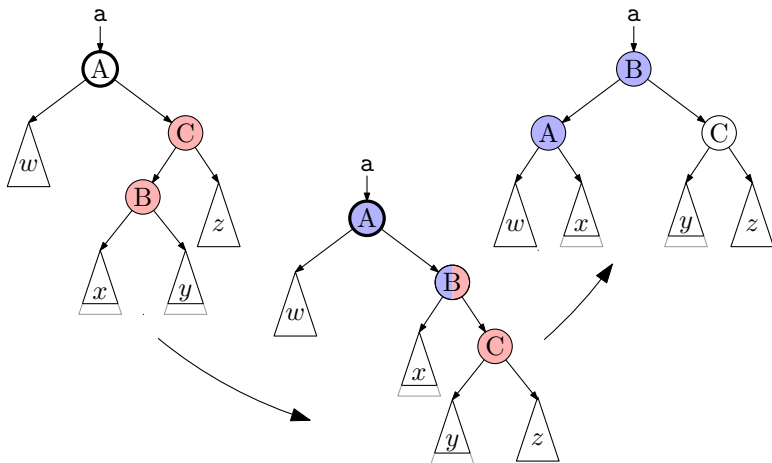
Insert(18)



Case RL: rotateRightLeft



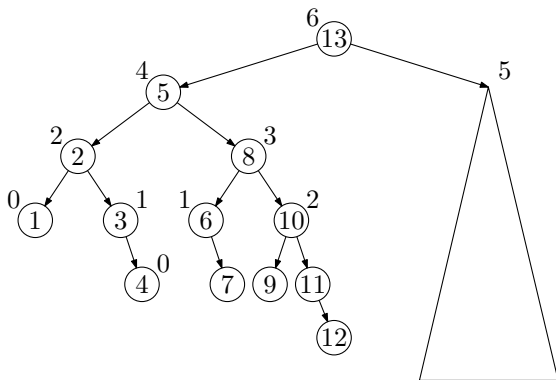
Double Rotation Code



```
void rotateRightLeft( Node *&a ) {  
    rotateRight(a->right);  
    rotateLeft(a);  
}
```

Remove

1. Remove as for general binary search tree (i.e. reduce the problem to removing a node with 0 or 1 child).
2. Go up tree from removed node searching for imbalance (and fixing heights).
3. Fix **all** unbalanced ancestors (bottom-up)



Thinking about AVL trees

Observations

- ▶ AVL trees are binary search trees that allow only slight imbalance
- ▶ Worst-case $O(\log n)$ time for find, insert, and remove
- ▶ Elements (even siblings) may be scattered in memory

Realities

- ▶ For large data sets, disk accesses dominate runtime

Could we have perfect balance if we relax binary tree restriction?