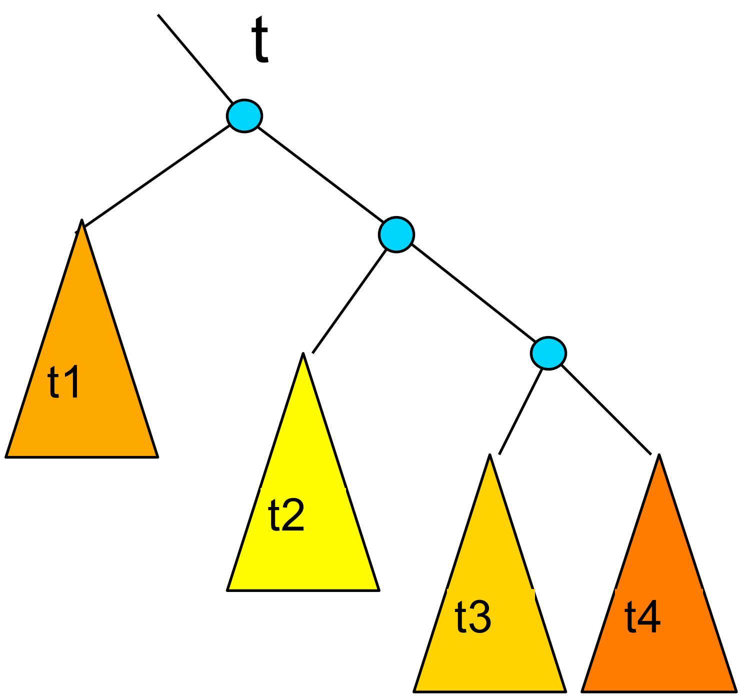
Announcements

MP5 available, due 10/30, 11:59p. EC due 10/23, 11:59p.

Exam 2: 11/3, 7-10p, rooms TBA

http://www.qmatica.com/DataStructures/Trees/AVL/AVLTree.html

AVL trees: diagnosing the correct rotation



Given: an insertion occurs to the right of t and an imbalance is detected at t…

Then: balance factor at t is _____, and *some* kind of ____________ rotation about t rebalances the tree.

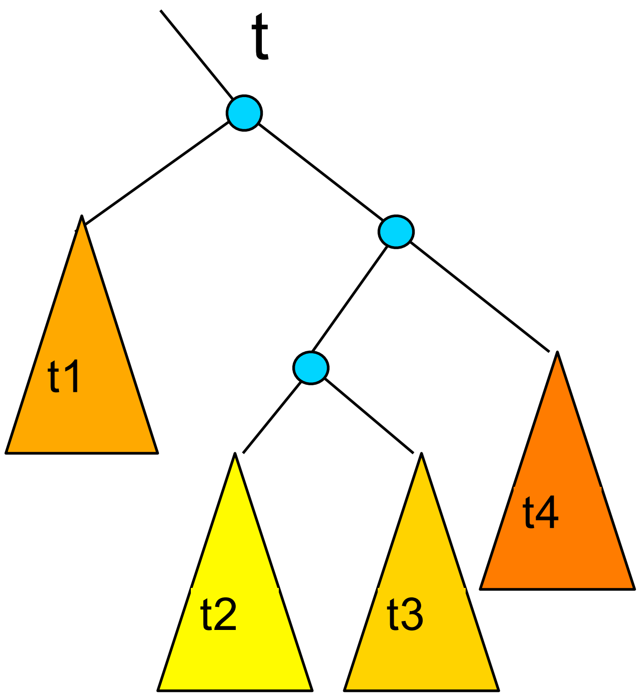
Further: the insertion occurs in t3 or t4

Then: balance factor at t->right is _____,

-1 0 1

And left rotation at t rebalances the tree

AVL trees: diagnosing the correct rotation



Given: an insertion occurs to the right of t and an imbalance is detected at t…

Then: balance factor at t is ____, and *some* kind of ___________ rotation about t rebalances the tree.

Further: the insertion occurs in t2 or t3

Then: balance factor at t->right is ____.

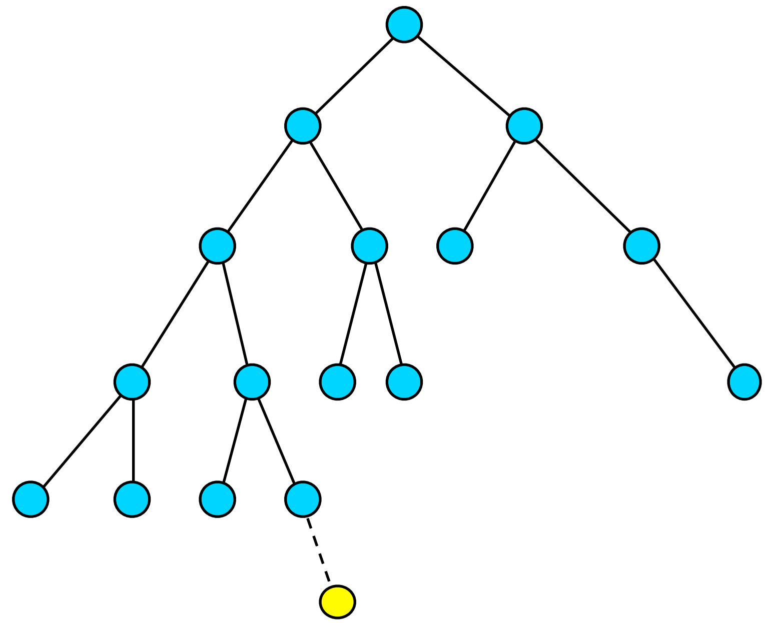
-1 0 1

And a right rotation about t->right, followed by a left rotation about t rebalances the tree.

(rightLeft double rotation)

AVL trees:

```
struct treeNode {
   T key;
   int height;
   treeNode * left;
   treeNode * right;
};
```



Insert:

insert at proper place

check for imbalance

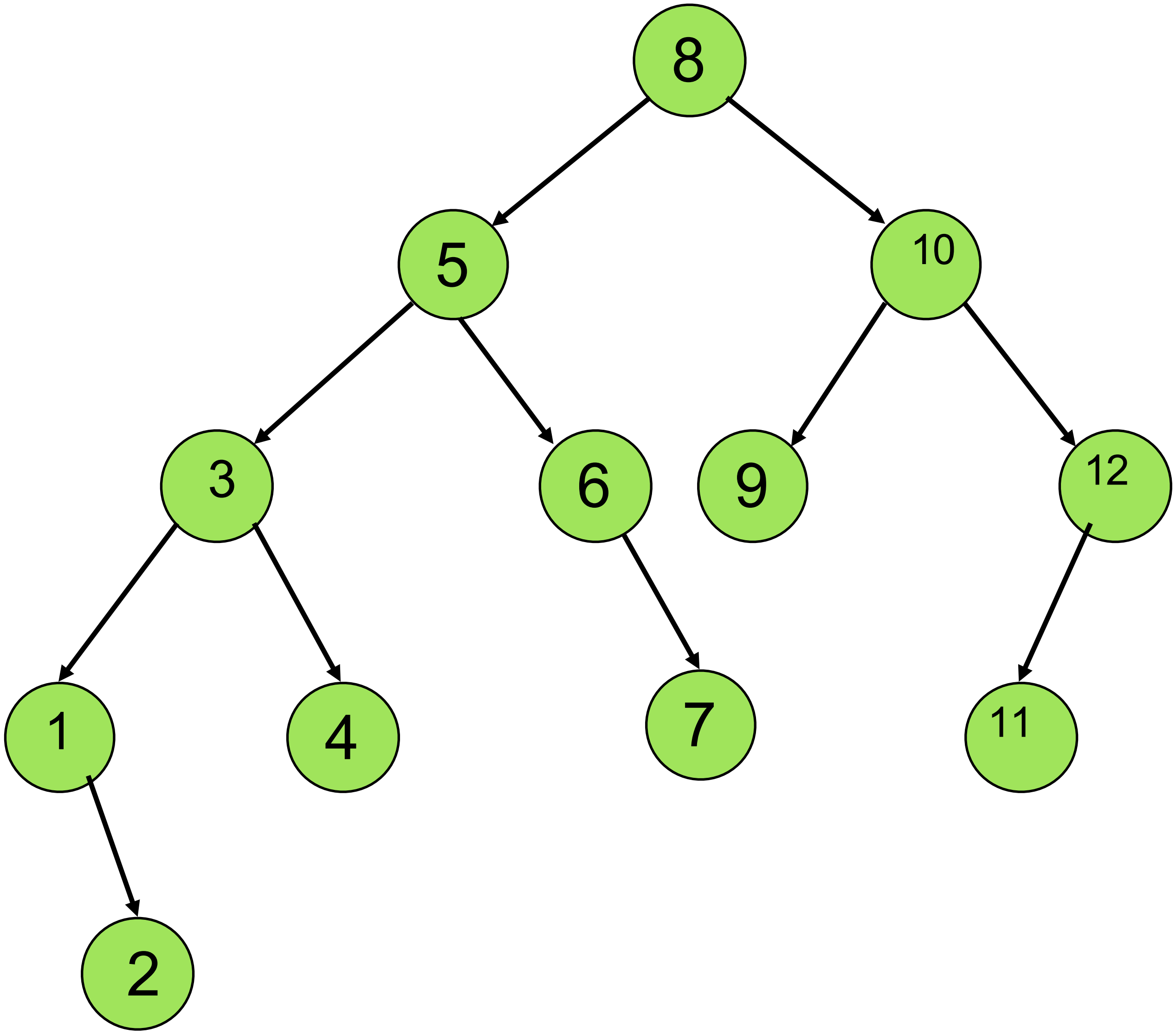
rotate if necessary

update height

AVL tree insertions:

```
template <class T>
void AVLTree<T>::insert(const T & x, treeNode<T> * & t ){

   if( t == NULL ) t = new treeNode<T>( x, 0, NULL, NULL);

   else if( x < t->key ){

      insert( x, t->left );
      if( balanceFactor(t) == -2 )
         if( balanceFactor(t->left) == -1 )
            rotate_____________( t );
         else
            rotate_____________( t );
   }
   else if( x > t->key ){

      insert( x, t->right );
      if( balanceFactor(t) == 2 )
         if( balanceFactor(t->right) == 1 )
            rotate_____________( t );
         else
            rotate_____________( t );
      }
   t->height=max(height(t->left ), height(t->right))+ 1;}
```

AVL tree removal:



AVL tree analysis:

Since running times for Insert, Remove and Find are O(h), we'll argue that h = O(log n).

- Defn of big-O:

- Draw two pictures to help us in our reasoning:

- Putting an upper bound on the height for a tree of n nodes is the same as putting a lower bound on the number of nodes in a tree of height h.

AVL tree analysis:

Putting an upper bound on the height for a tree of n nodes is the same as putting a lower bound on the number of nodes in a tree of height h.

- Define N(h):

- Find a recurrence for N(h):

- We simplify the recurrence:

- Solve the recurrence: (guess a closed form)

AVL tree analysis: prove your guess is correct.

Thm: An AVL tree of height h has at least $2^{h/2}$ nodes, _________.

Consider an arbitrary AVL tree, and let h denote its height.

Case 1: ______

Case 2: ______

Case 3: ______ *then, by an Inductive Hypothesis that says*

__________________________, *and since*

__________________________, *we know that*

__________________________.

Punchline: