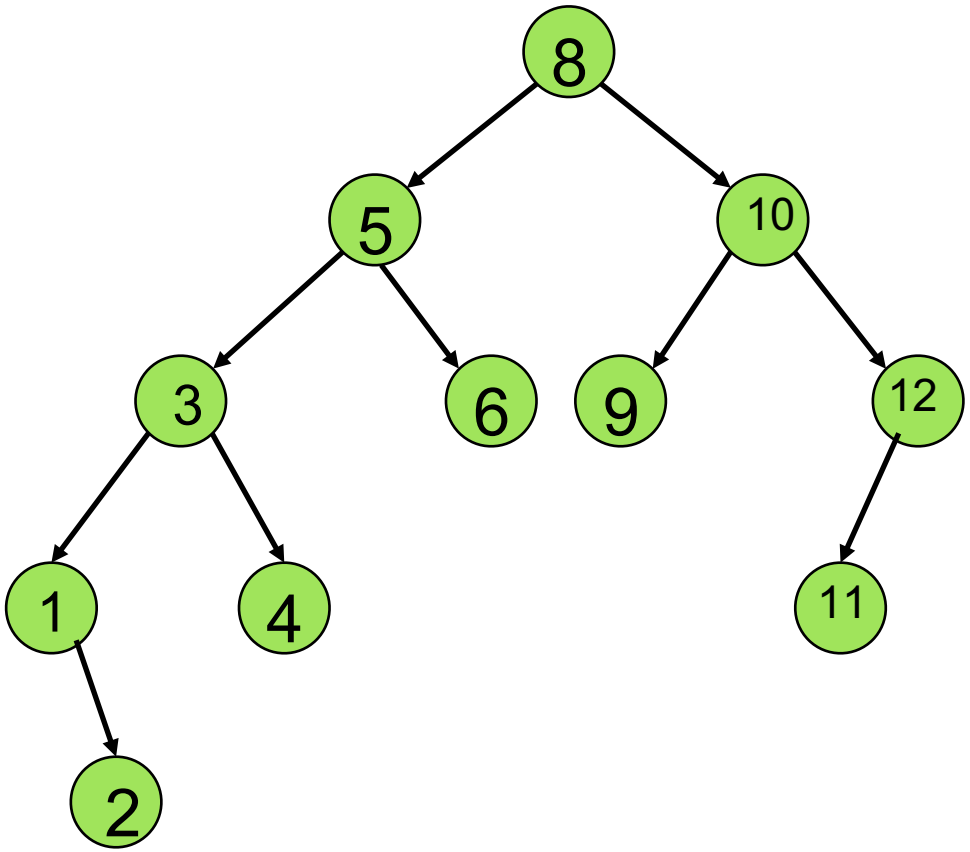
Announcements

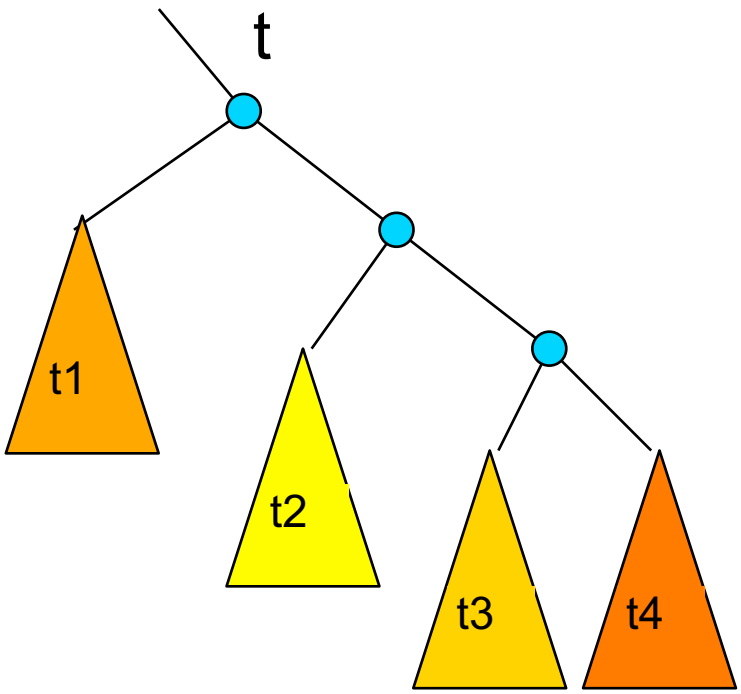
MP5 available, due 11/1, 11:59p. EC due 10/25, 11:59p.

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

Maintaining height upon a rotation:



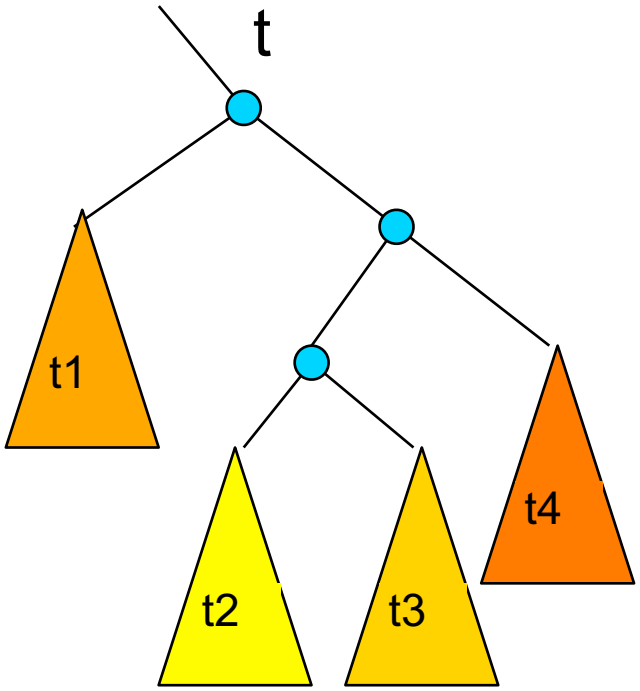
AVL trees: rotations (identifying the need)



if an insertion was in subtrees t3 or t4, and if an imbalance is detected at t, then a _____ rotation about t rebalances the tree.

We gauge this by noting that the balance factor at t->right is _____

AVL trees: rotations (identifying the need)

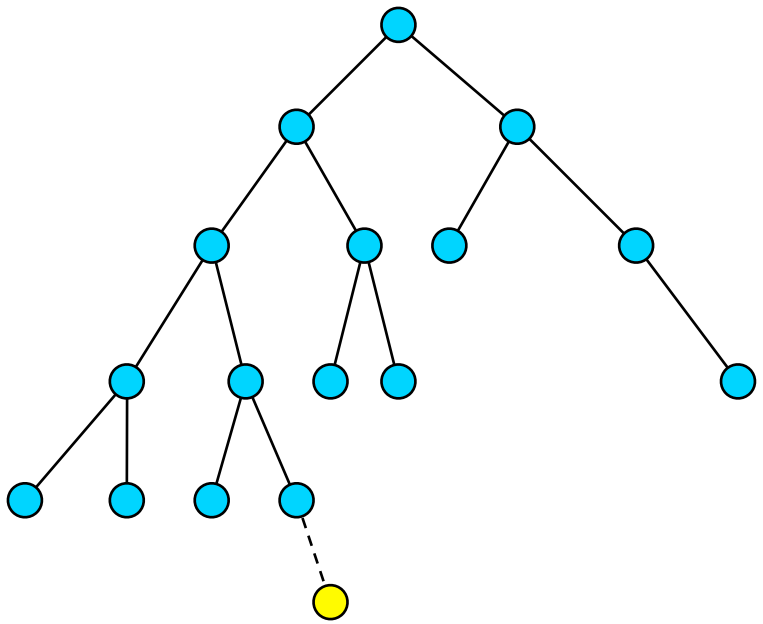


If an insertion was in subtrees t2 or t3, and if an imbalance is detected at t, then a _____ rotation about t rebalances the tree.

We gauge this by noting that the balance factor at t->right is _____

AVL trees:

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



Insert:

insert at proper place

check for imbalance

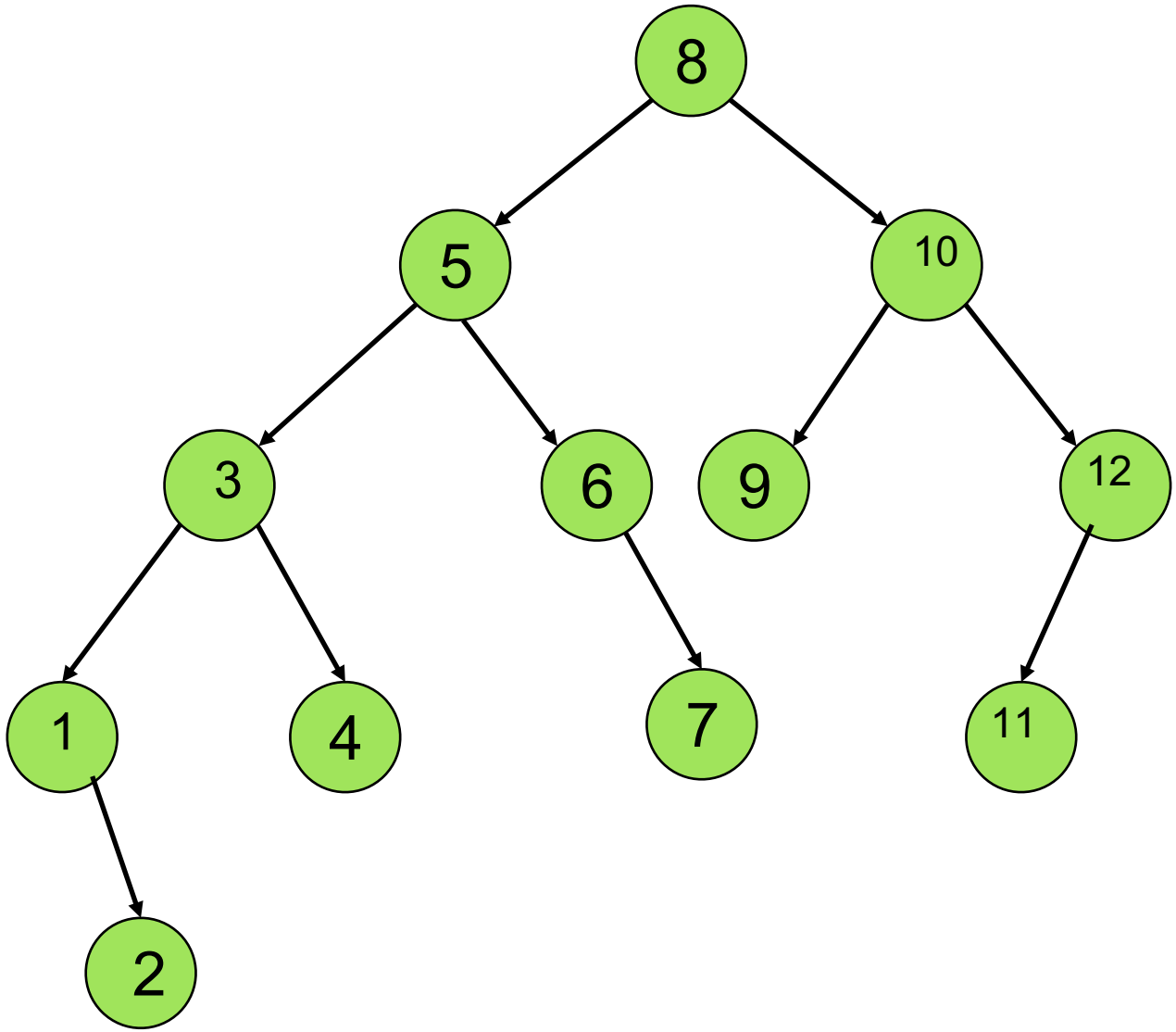
rotate if necessary

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 );
      int balance = height(t->right)-height(t->left);
      int leftBalance = height(t->left->right)-height(t->left->left);
      if( balance == -2 )
         if( leftBalance == -1 )
            rotate_____________( t );
         else
            rotate_____________( t );
   }
   else if( x > t->key ){
      insert( x, t->right );
      int balance = height(t->right)-height(t->left);
      int rightBalance = height(t->right->right)-height(t->right->left);
      if( balance == 2 )
         if( rightBalance == 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)