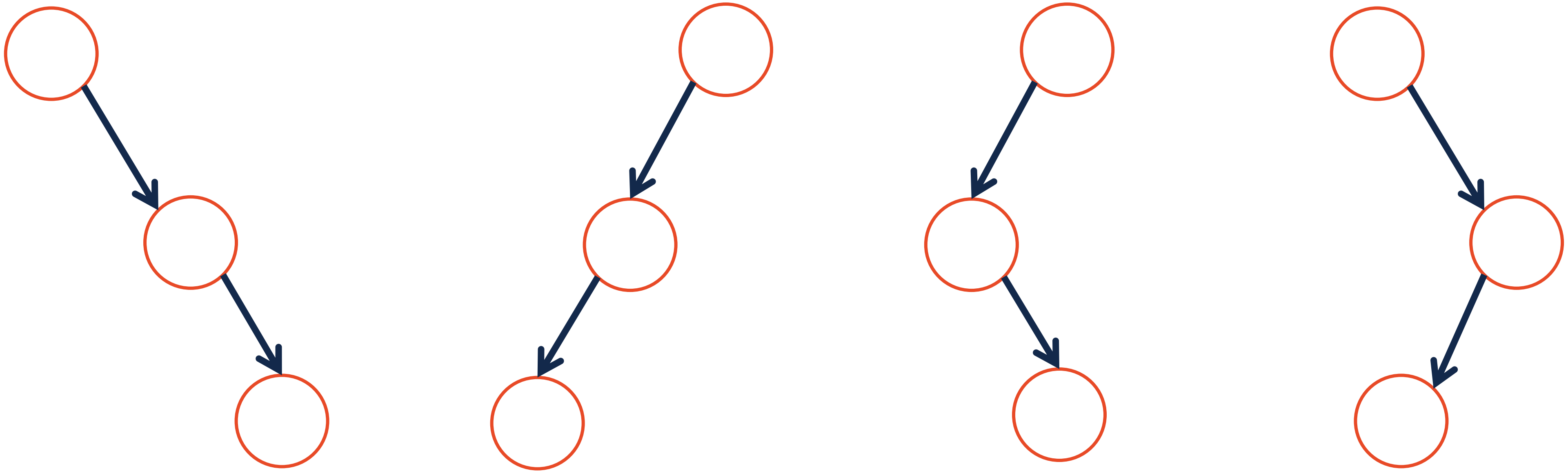
CS 225

Data Structures

Oct. 16 – AVL Operations

AVL Tree Rotations

Four templates for rotations:



Height-Balanced Tree

Height balance: $b = \text{height}(T_R) - \text{height}(T_L)$

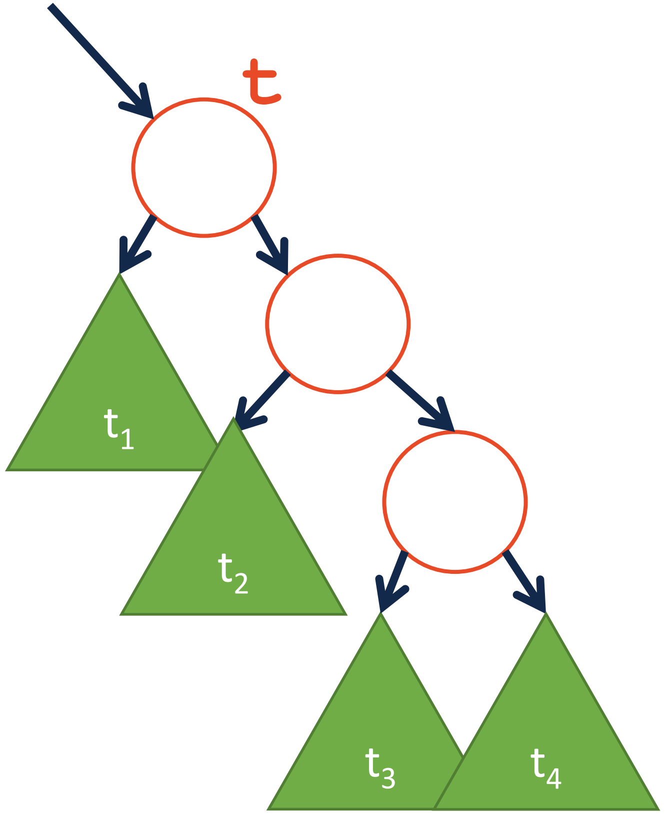
Exams

Current Exam: Exam 6 (Programming)

Next Week: Exam 7 (Theory)

- **Topics:**
  - **Trees:** Binary, Binary Search, AVL Rotations
  - **Iterators**
  - **Functors**
  - **MP 3**
  - **Huffman Encoding**
- **A study guide will be released in the next few days**

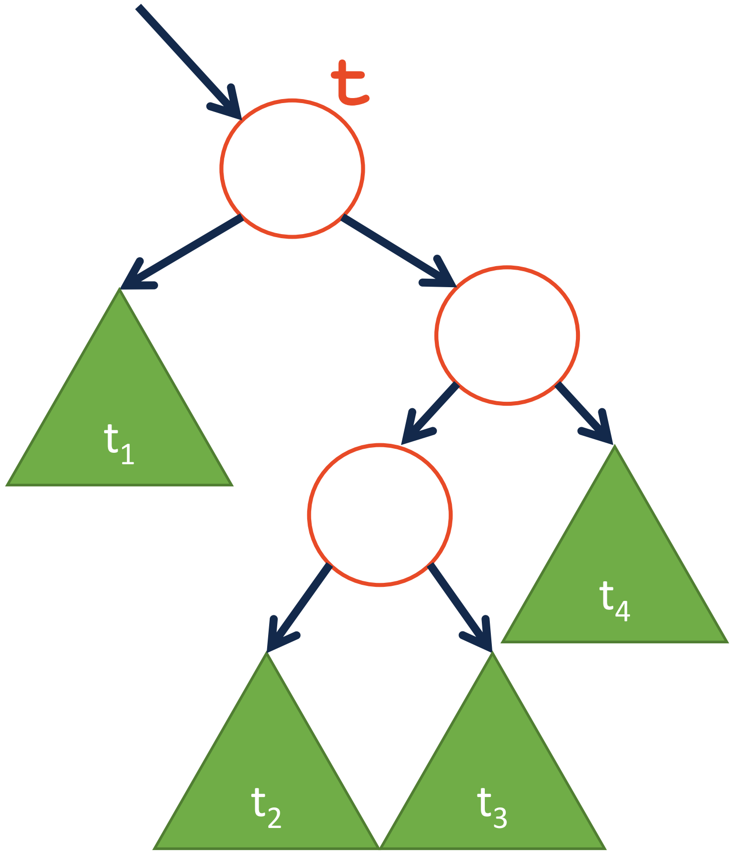
Finding the Rotation



Theorem:
If an insertion occurred in subtrees **t_3** or **t_4** and a subtree was detected at **t**, then a ____________ rotation about **t** restores the balance of the tree.

We gauge this by noting the balance factor of **t->right** is _______.

Finding the Rotation



Theorem:
If an insertion occurred in subtrees **t_2** or **t_3** and a subtree was detected at **t**, then a ____________ rotation about **t** restores the balance of the tree.

We gauge this by noting the balance factor of **t->right** is _______.

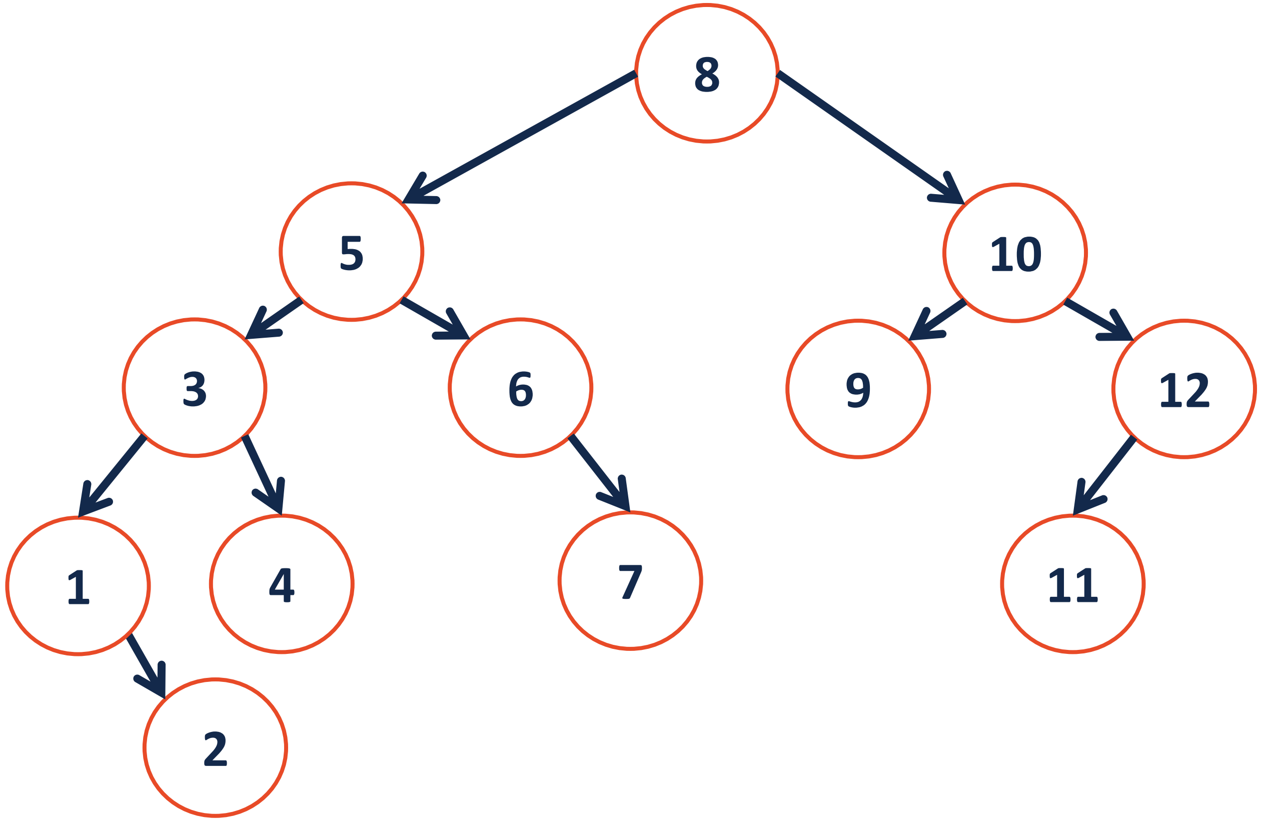
Insertion into an AVL Tree

`_insert(6.5)`

Insert (pseudo code):

```
1: Insert at proper place
2: Check for imbalance
3: Rotate, if necessary
4: Update height
```

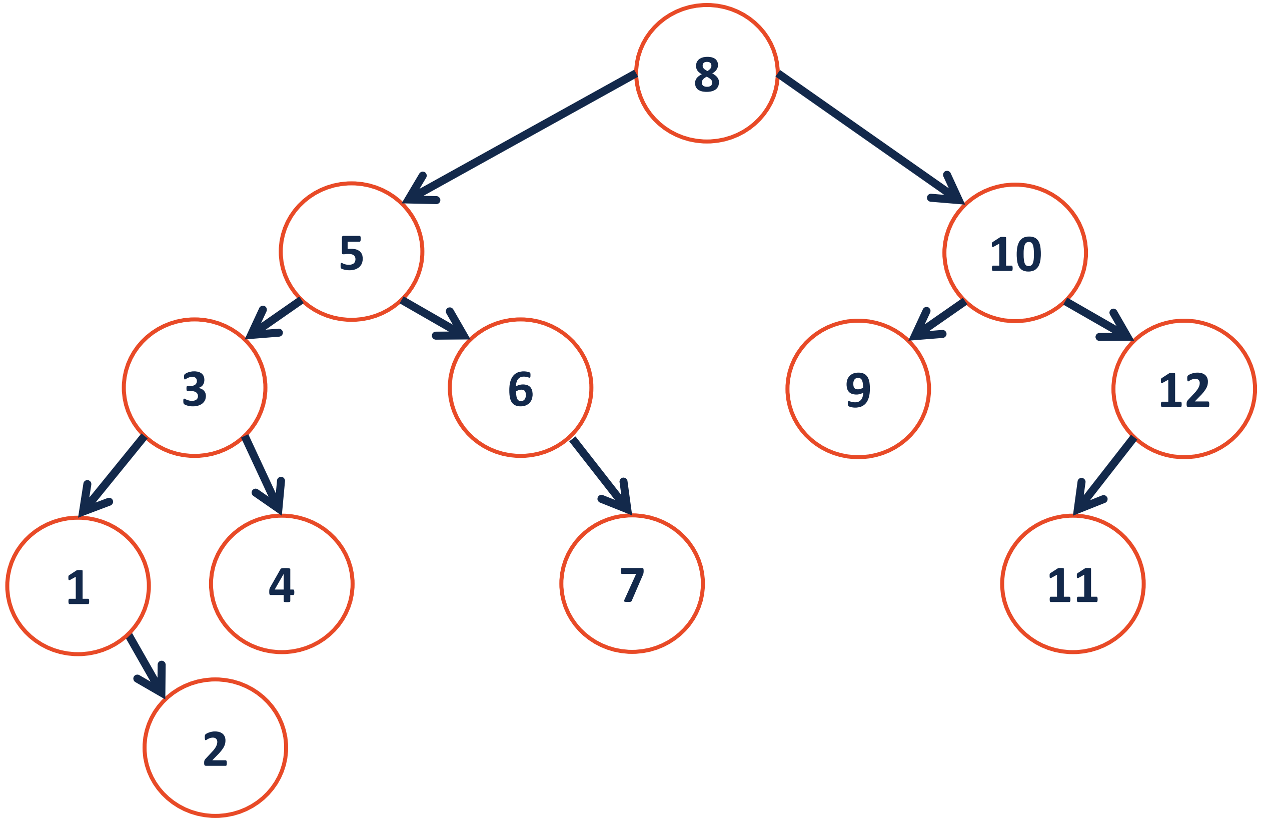
```
struct TreeNode {
  T key;
  unsigned height;
  TreeNode *left;
  TreeNode *right;
};
```



```
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 = 1 + max(height(t->left), height(t->right));
}
```

Height-Balanced Tree

Height balance: $b = \text{height}(T_R) - \text{height}(T_L)$


8
5
10
3
6
9
12
1
4
7
11
2

AVL Tree Analysis

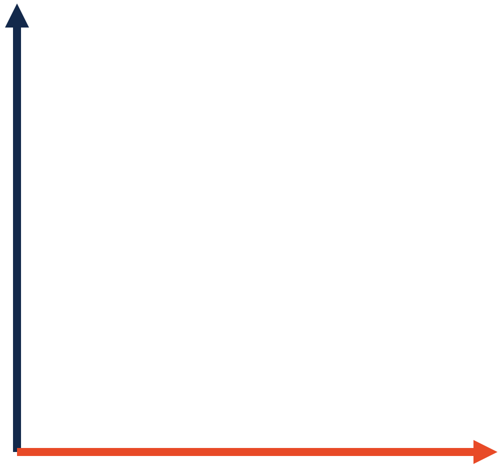
We know: insert, remove and find runs in: ____________.

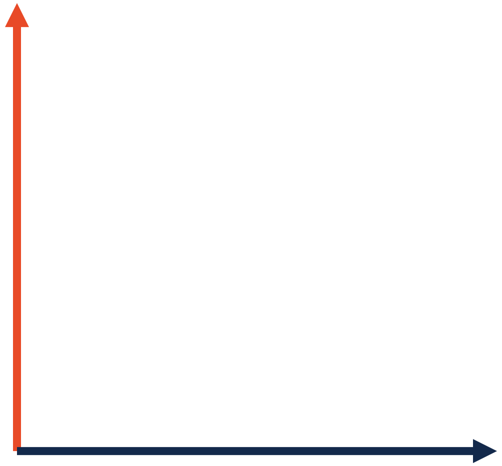
We will argue that: h = ___________.

AVL Tree Analysis

Definition of big-O:

…or, with pictures:





CS 225 – Things To Be Doing

Exam 6 (Programming, Lists/Trees) is ongoing!

More Info: https://courses.engr.illinois.edu/cs225/fa2017/exams/

MP4: Available now!

Due: Monday, Oct. 23 at 11:59pm

Labs

New lab on Wednesday

POTD

Every Monday-Friday – *Worth +1 Extra Credit /problem (up to +40 total)*