Data Structures

Balanced Binary Search Trees

CS 225 September 22, 2023

Brad Solomon & G Carl Evans

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Department of Computer Science

Learning Objectives

Discuss the big picture problem with BSTs

Introduce the self-balancing BST

BST Analysis

Every operation on a BST depends on the **height** of the tree.

… how do we relate $O(h)$ to n, the size of our dataset?

BST Analysis

What is the **max** number of nodes in a tree of height h ?

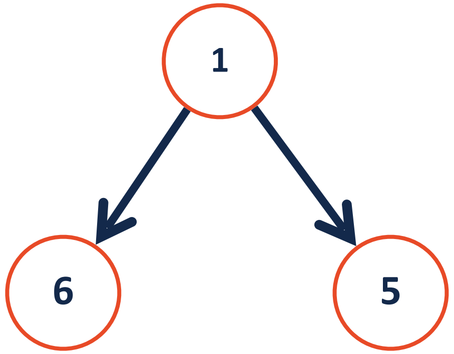
BST Analysis

What is the **min** number of nodes in a tree of height h ?

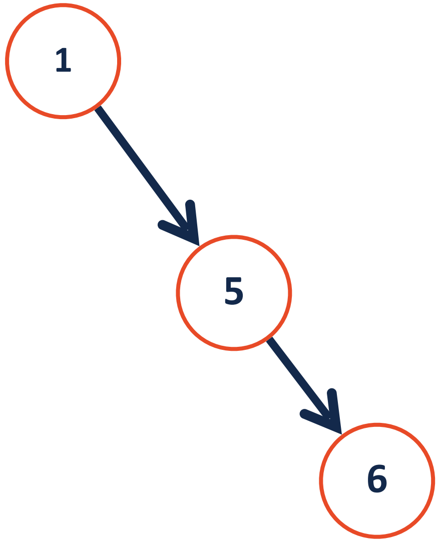
BST Analysis

A BST of n nodes has a height between:

Lower-bound: $O(\log n)$

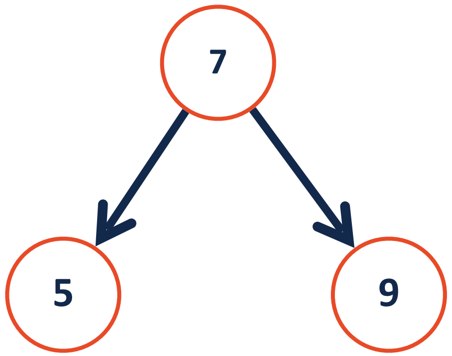


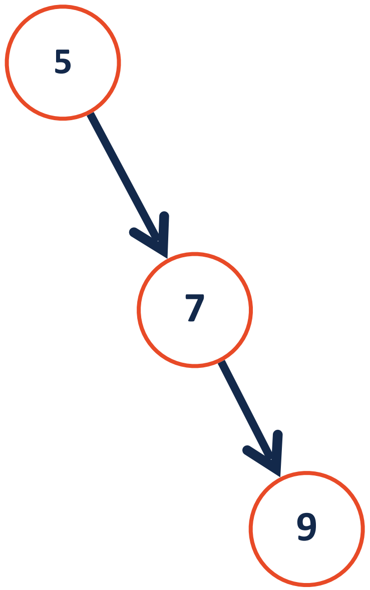
Upper-bound: $O(n)$



Height-Balanced Tree

What tree is better?





Height balance: $b = height(T_R) - height(T_L)$

A tree is "balanced" if:

BST Rotations (The AVL Tree)

We can adjust the BST structure by performing **rotations**.

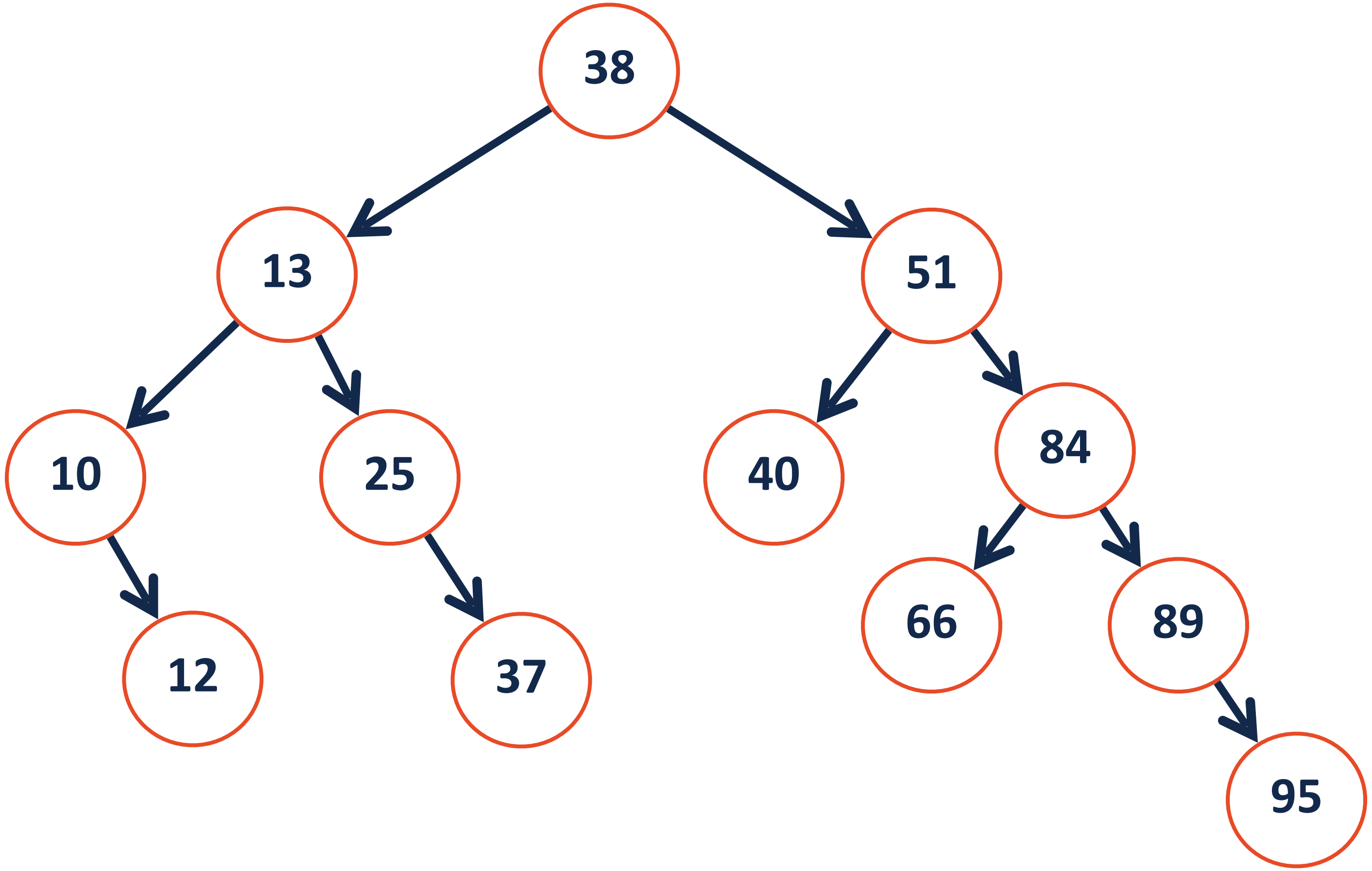
These rotations:

1.

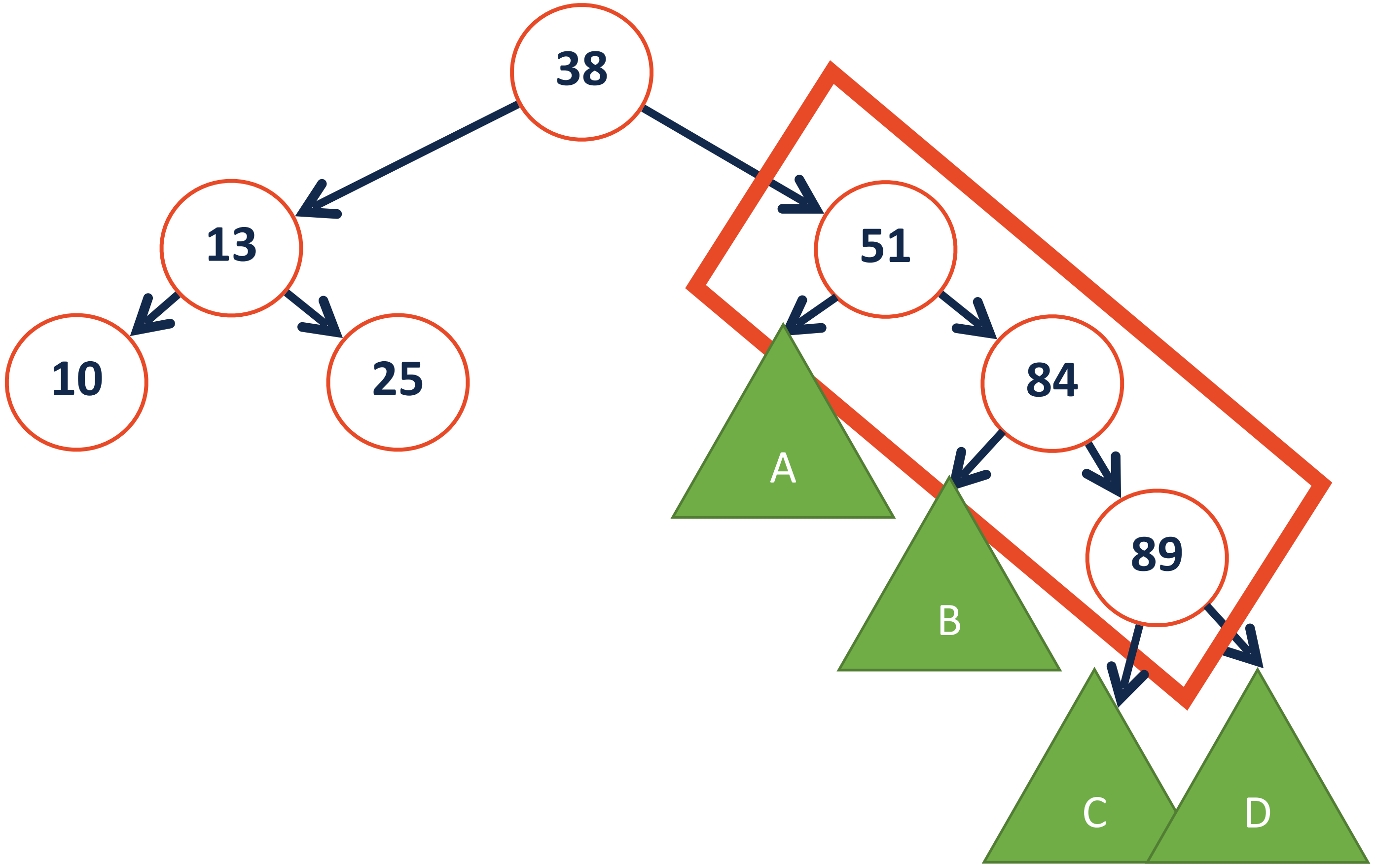
2.

BST Rotations (The AVL Tree)

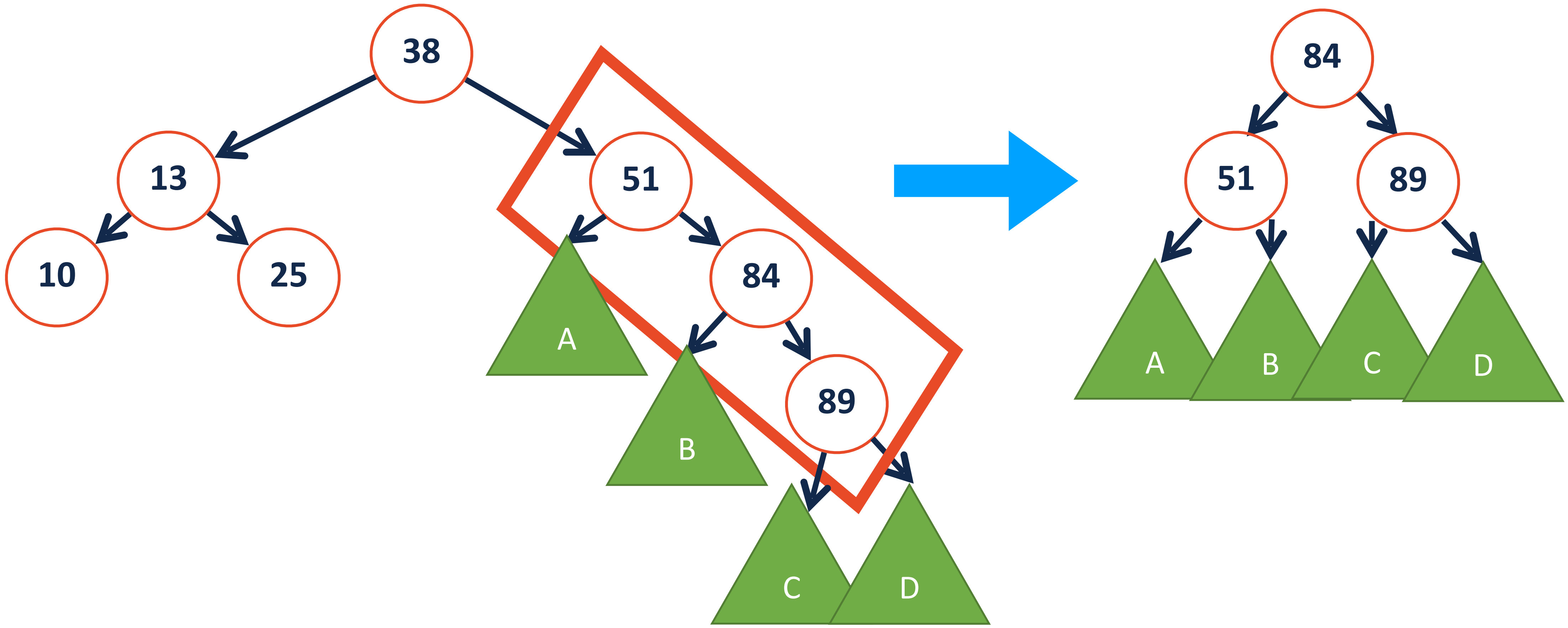
We can adjust the BST structure by performing **rotations**.



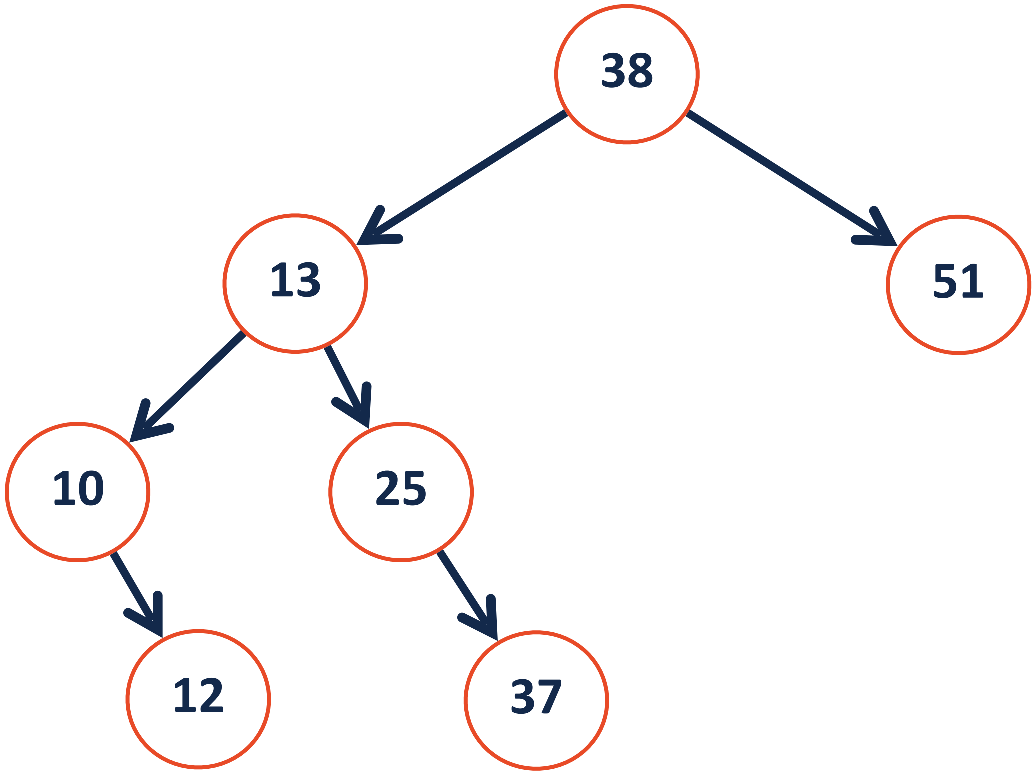
Left Rotation



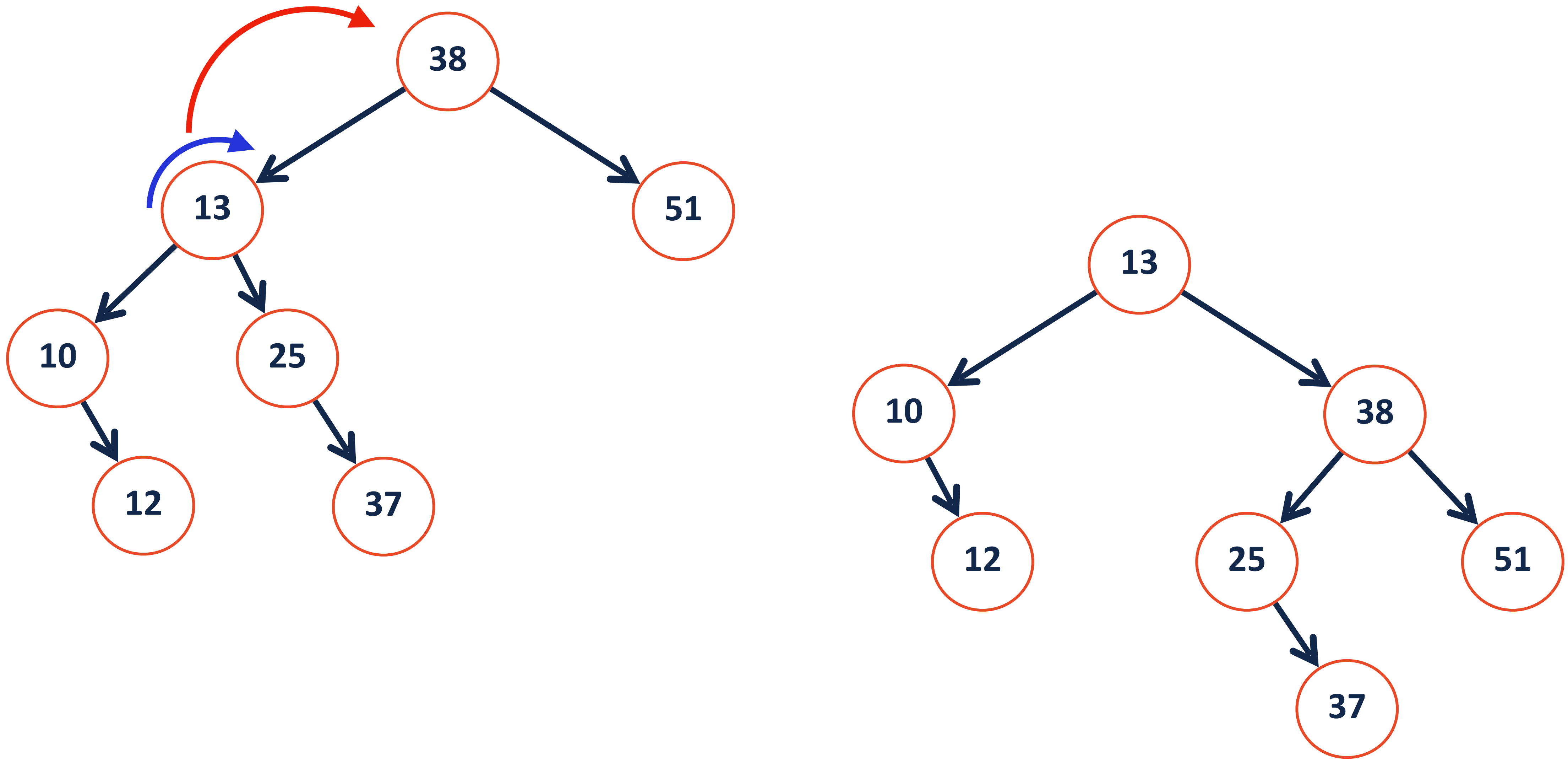
Left Rotation



Right Rotation



Right Rotation

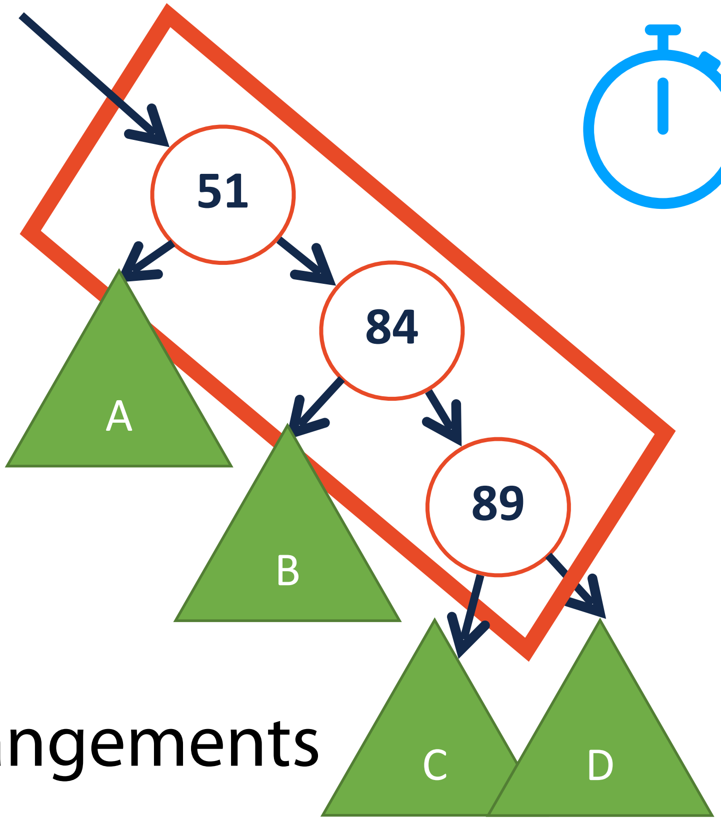


Coding AVL Rotations

Two ways of visualizing:

1) Think of an arrow 'rotating' around the center

2) Recognize that there's a concrete order for rearrangements



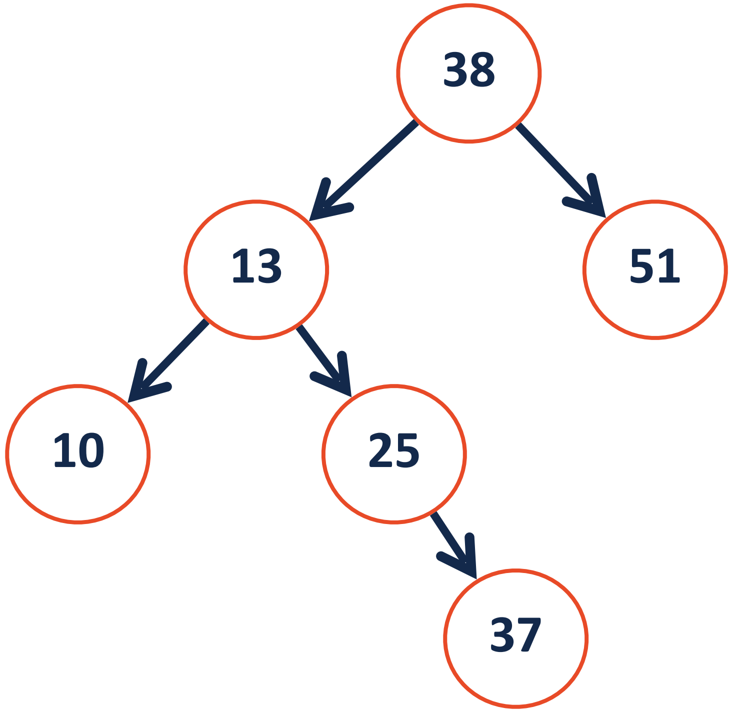
Ex: Unbalanced at current (root) node and need to *rotateLeft*?

Replace current (root) node with it's right child.

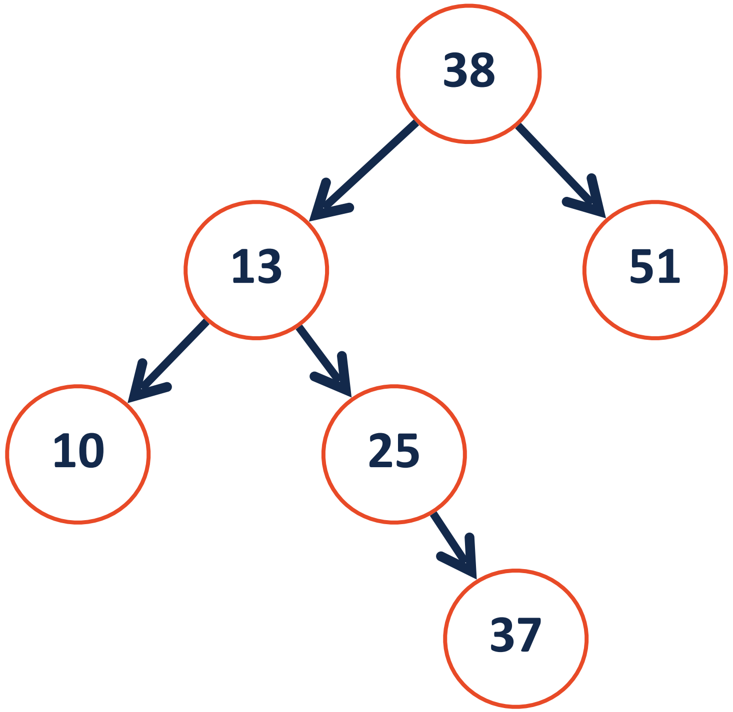
Set the right child's left child to be the current node's right

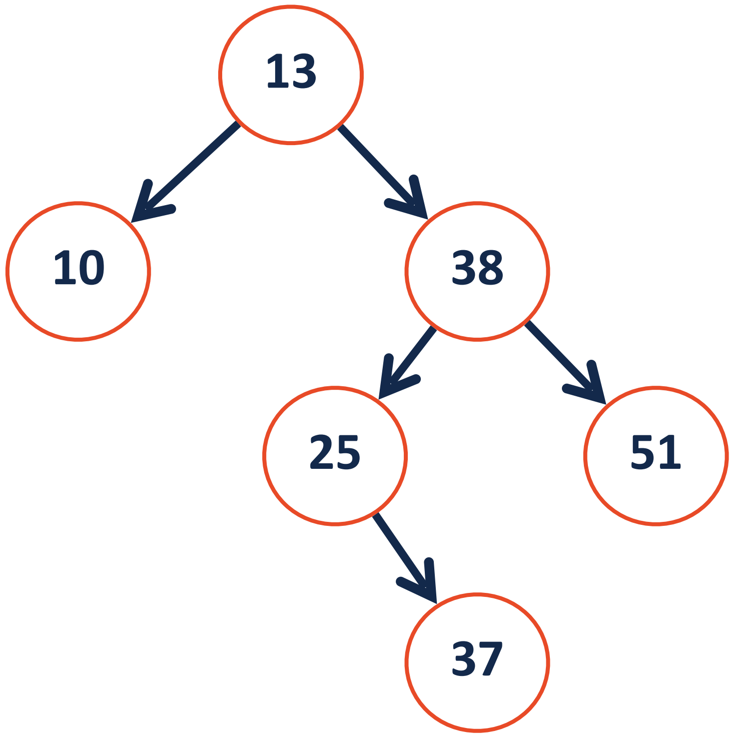
Make the current node the right child's left child

AVL Rotation Practice



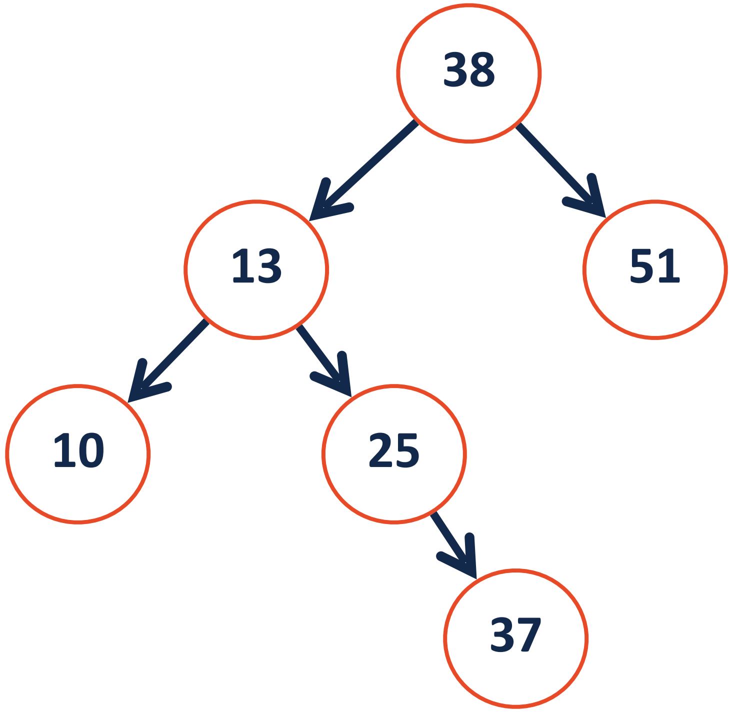
AVL Rotation Practice



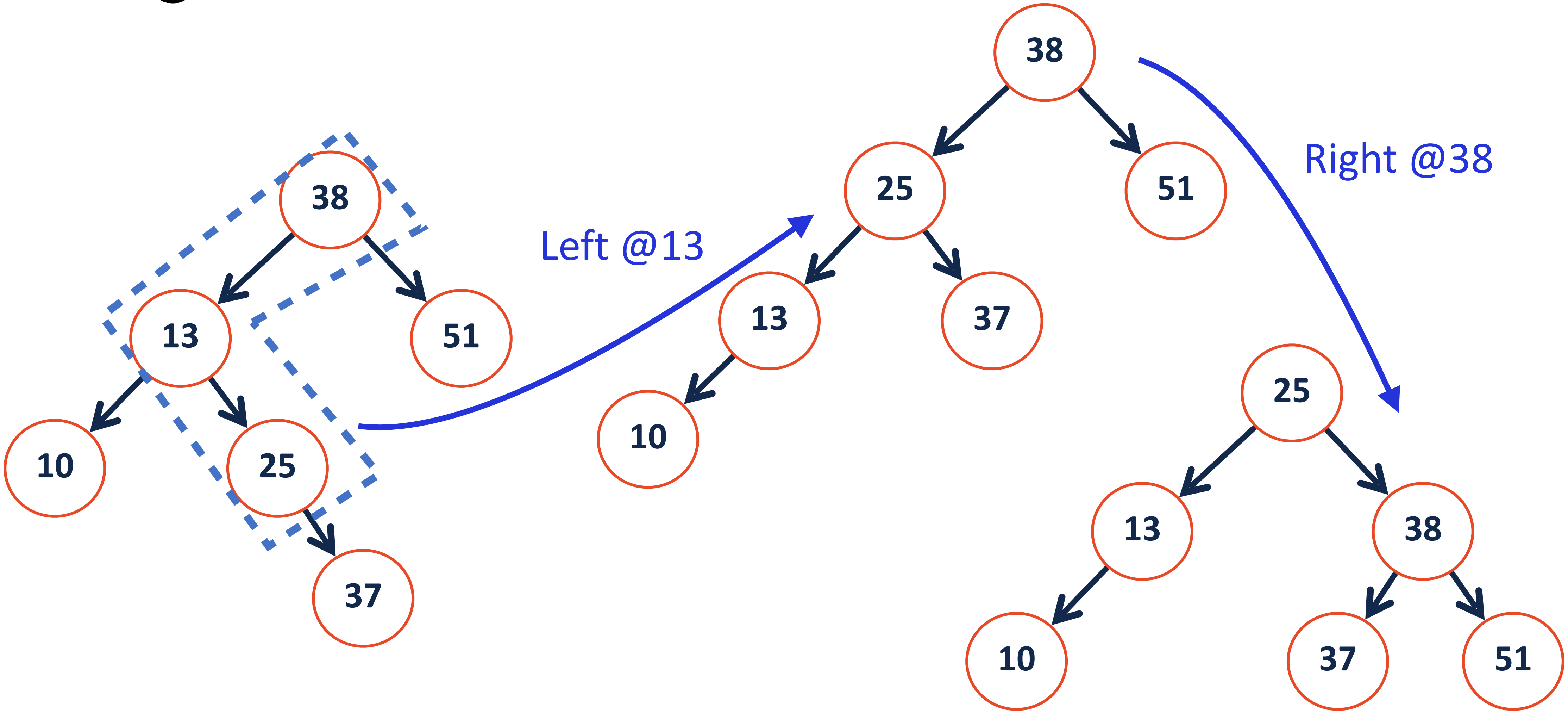


Somethings not quite right…

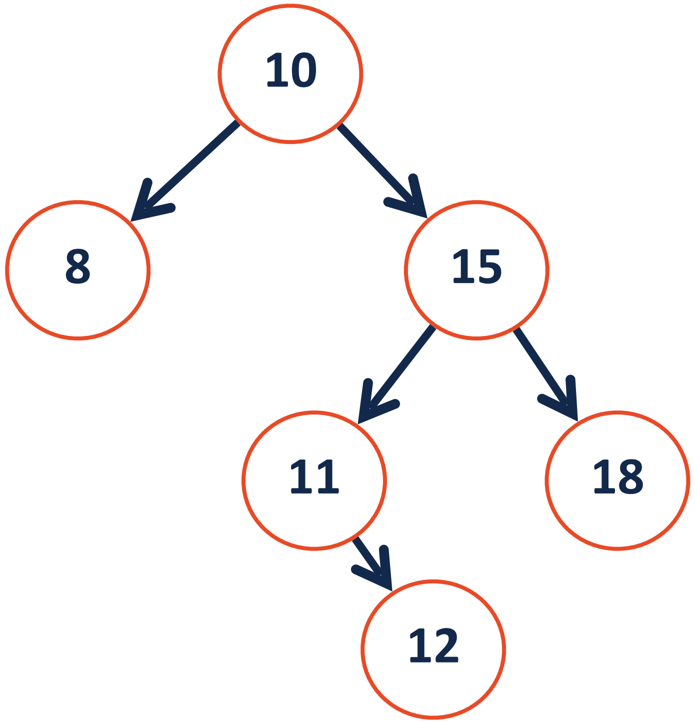
LeftRight Rotation



LeftRight Rotation

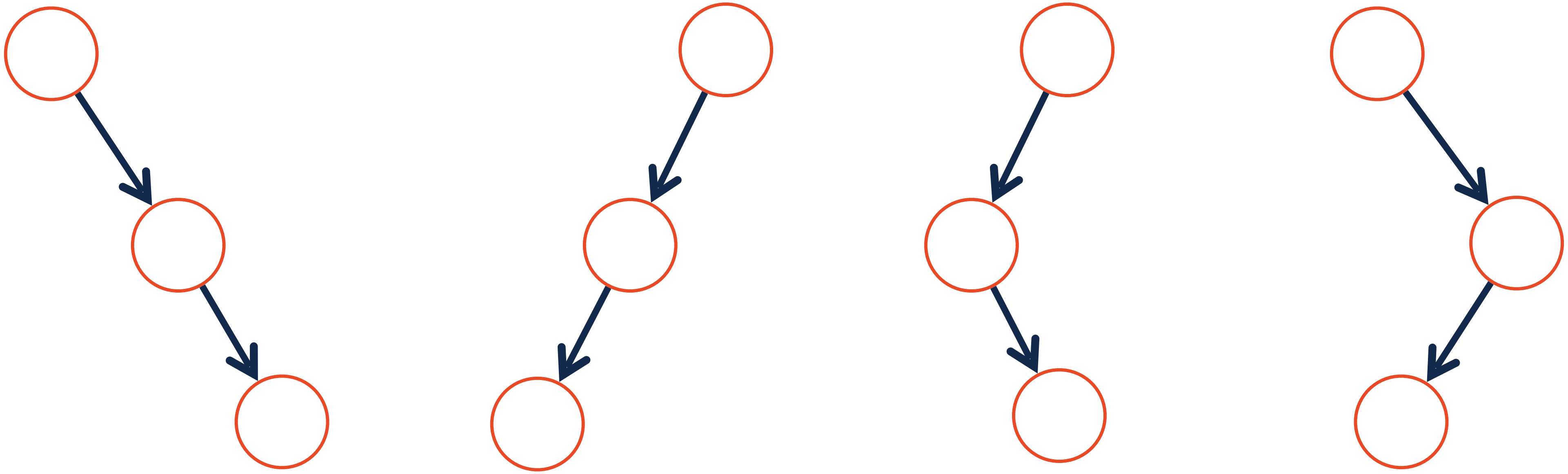


RightLeft Rotation



AVL Rotations

Four kinds of rotations:



AVL Rotations

Four kinds of rotations: (L, R, LR, RL)

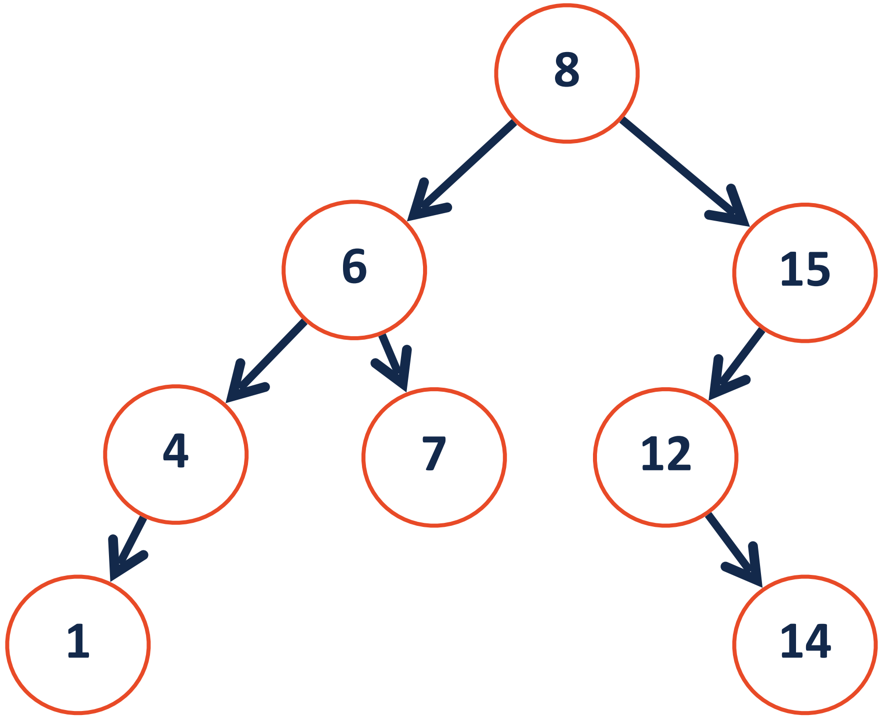
1. All rotations are local (subtrees are not impacted)

2. The running time of rotations are constant

3. The rotations maintain BST property

Goal:

AVL Rotation Practice



AVL vs BST ADT

The AVL tree is a modified binary search tree that rotates **when necessary**

How does the constraint on balance affect the core functions?

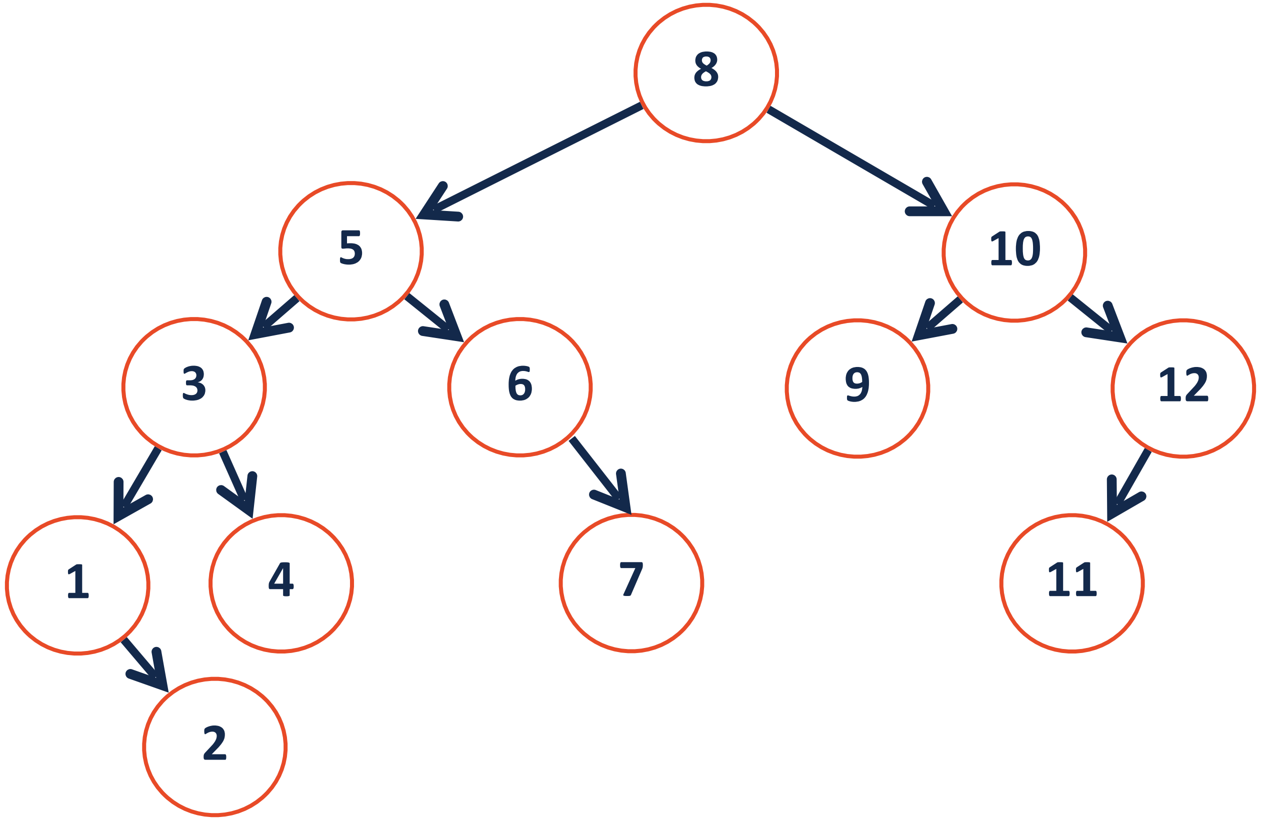
Find

Insert

Remove

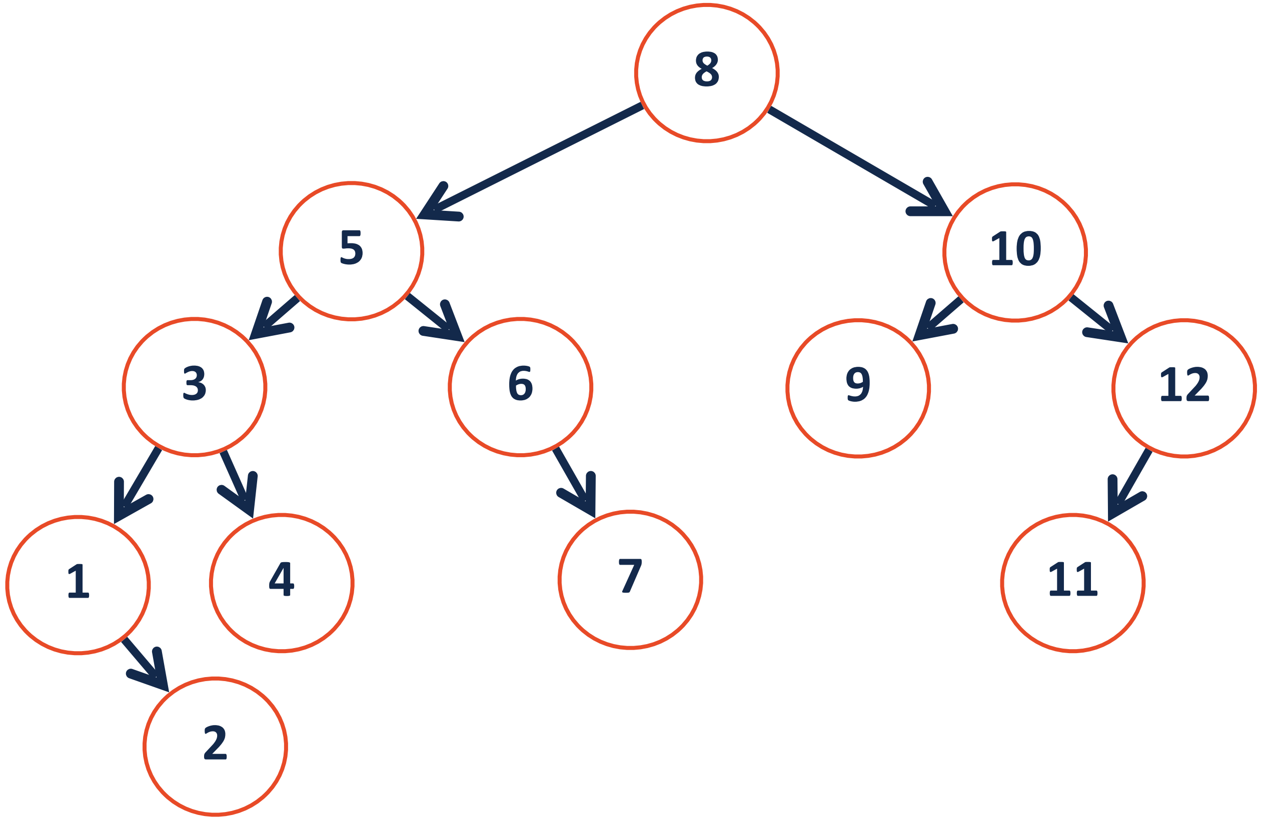
AVL Find

`_find(7)`



AVL Insertion

`_insert(6.5)`




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

AVL Insertion

`_insert(6.5)`

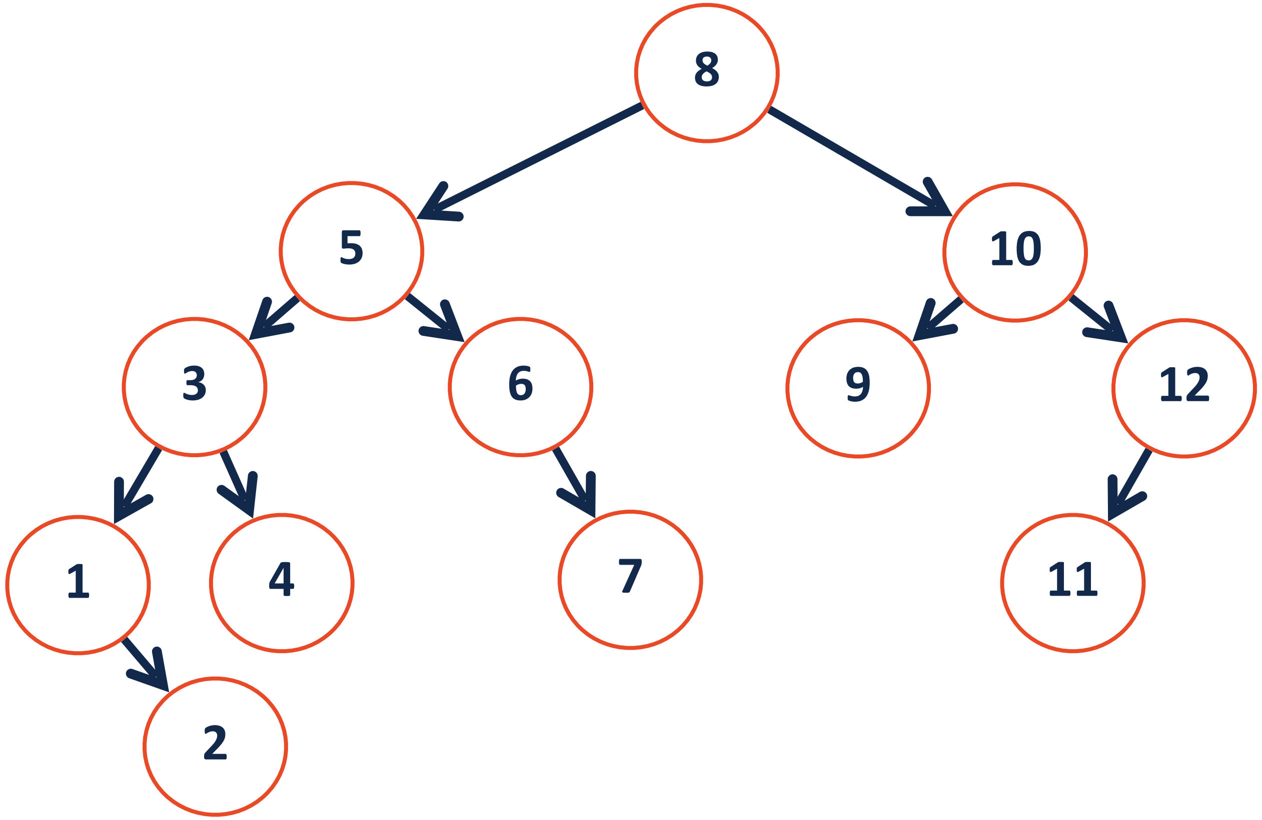
Insert (pseudo code):

1: Insert at proper place

2: Check for imbalance

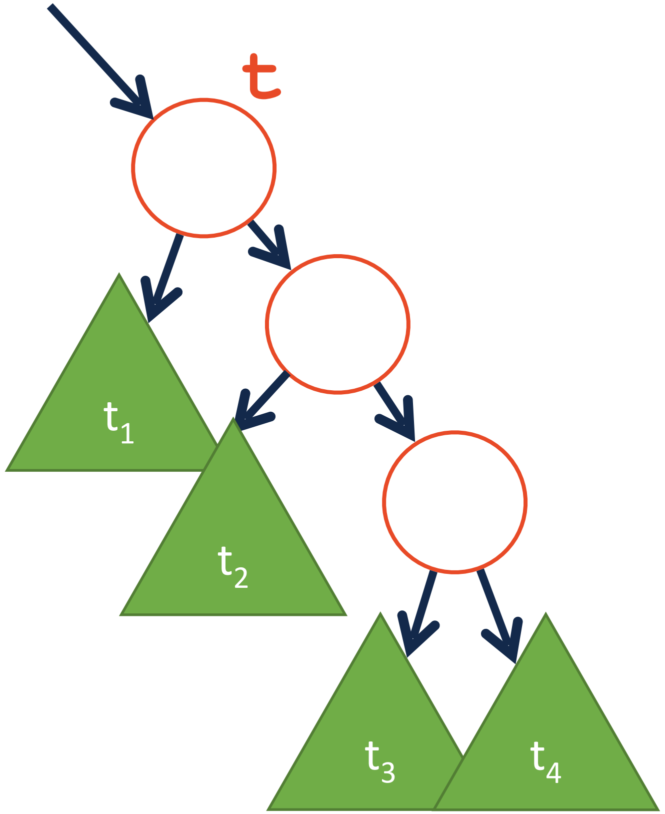
3: Rotate, if necessary

4: Update height




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

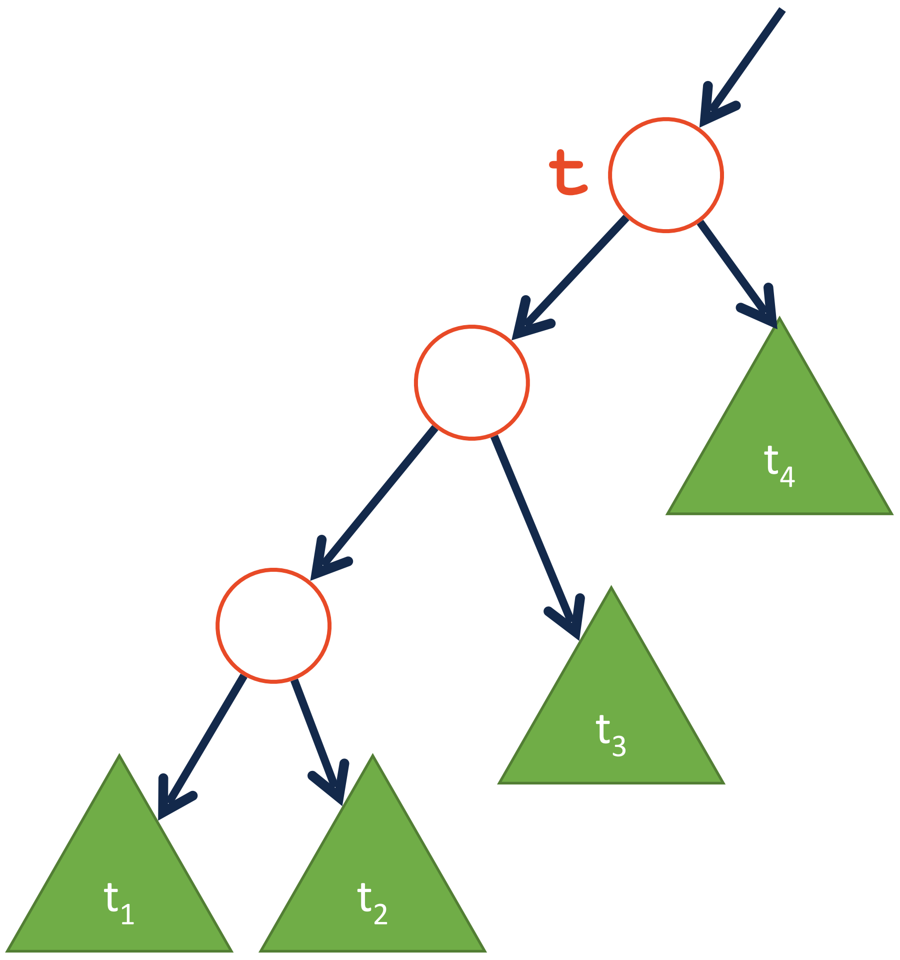
Rebalancing on insert



Theorem:
If an insertion occurred in subtrees $\mathbf{t_3}$ or $\mathbf{t_4}$ and an imbalance was first detected at **t**, then a _______________ rotation about **t** restores the balance of the tree.

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

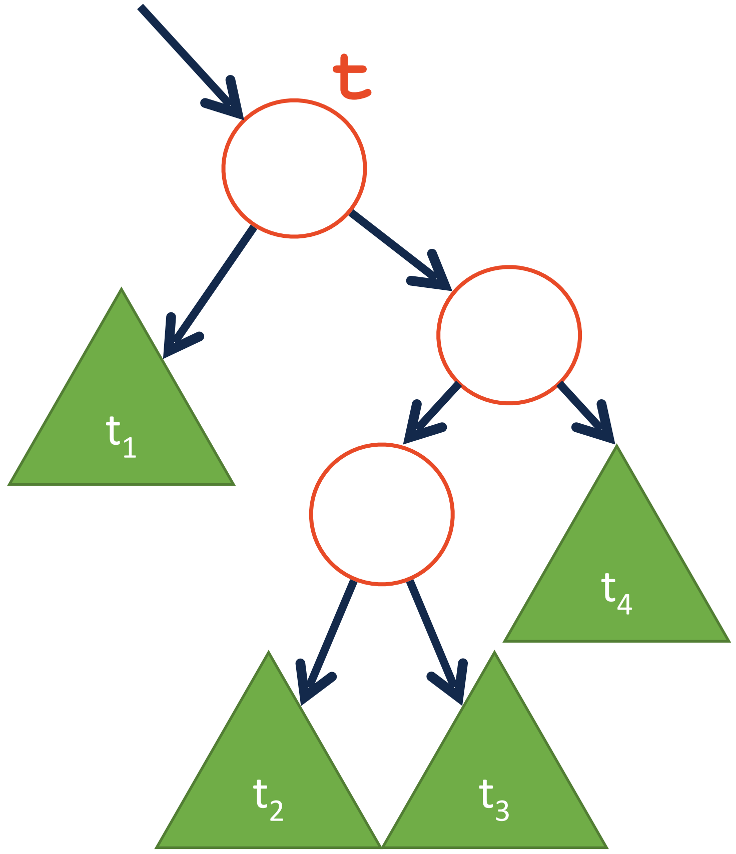
Rebalancing on insert



Theorem:
If an insertion occurred in subtrees $\mathbf{t_1}$ or $\mathbf{t_2}$ and an imbalance was first detected at **t**, then a _______________ rotation about **t** restores the balance of the tree.

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

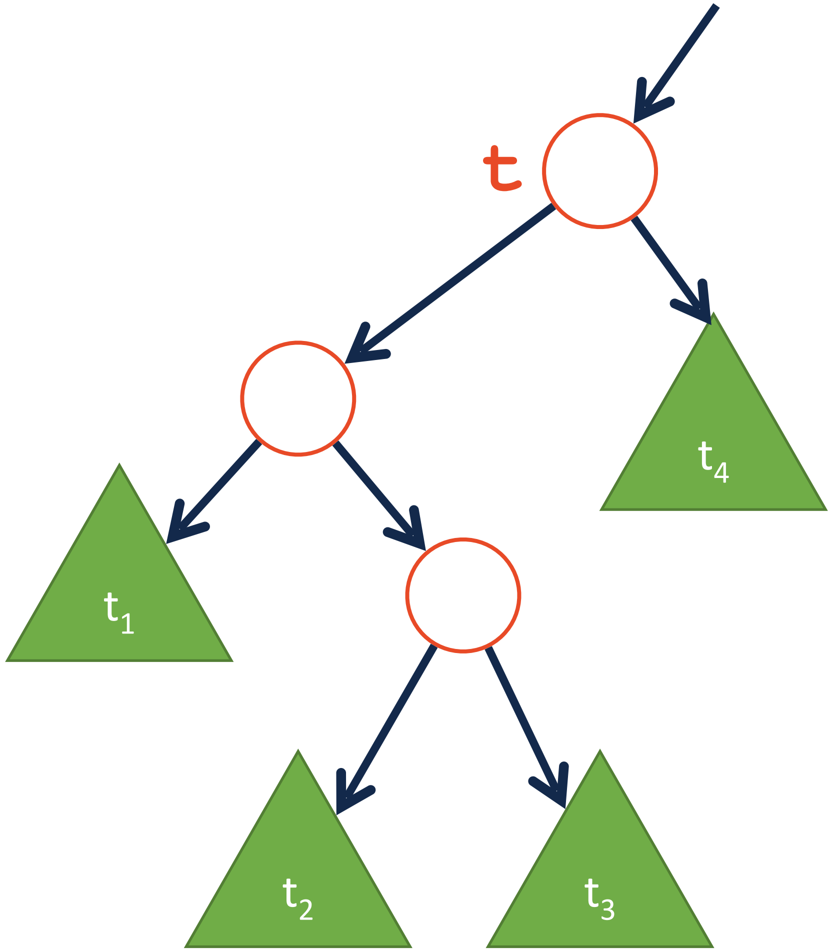
Rebalancing on insert



Theorem:
If an insertion occurred in subtrees $\mathbf{t_2}$ or $\mathbf{t_3}$ and an imbalance was first detected at **t**, then a _______________ rotation about **t** restores the balance of the tree.

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

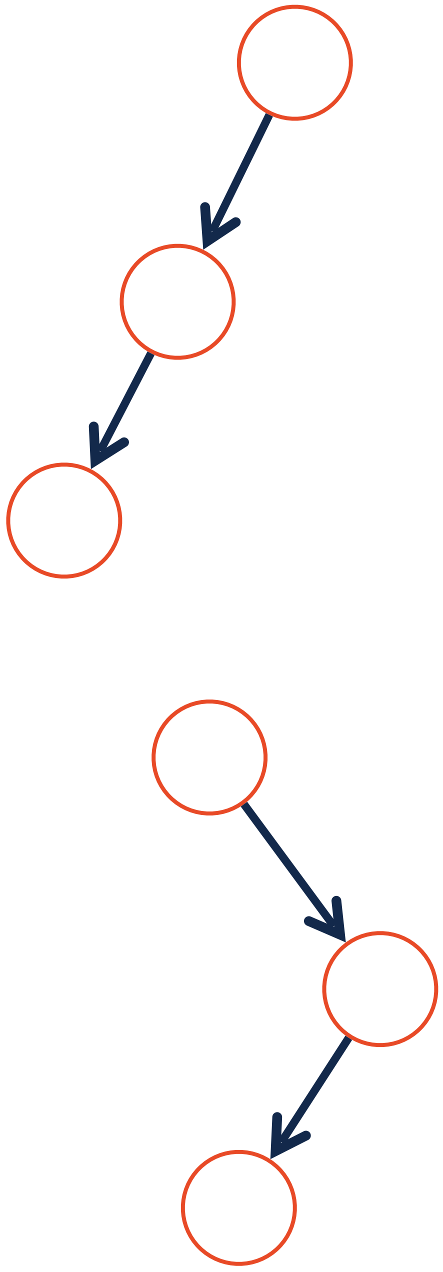
Rebalancing on insert



Theorem:
If an insertion occurred in subtrees $\mathbf{t_2}$ or $\mathbf{t_3}$ and an imbalance was first detected at **t**, then a ______________ rotation about **t** restores the balance of the tree.

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

Rebalancing on insert



AVL Insertion Practice

`_insert(14)`