Example 2: A Complex Rotation
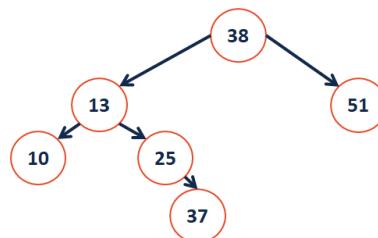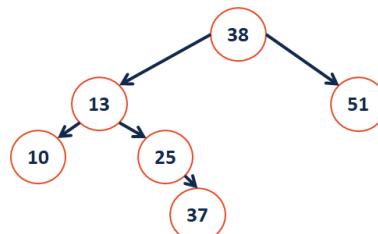
BST Rotation Summary:
1. Four kinds of rotations (L, R, LR, and RL)
2. All rotations are local
3. All rotations run in constant time, O(1)
4. BST property is maintained!

Overall Goal:

...and we call these trees:
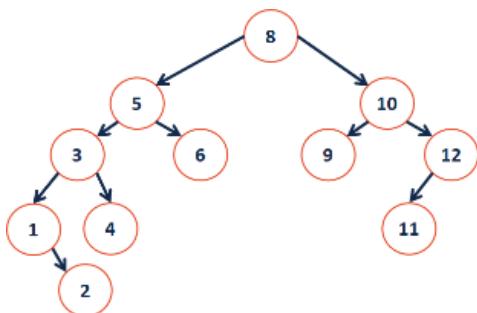
...additional property:


AVL Theorem #1: 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.
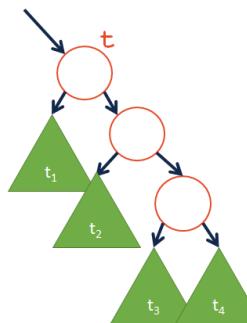
AVL Theorem #2: If an insertion occurred in subtrees t_2 or t_3 and a subtree was detected at **t**, then a _____ rotation about t restores the balance of the tree.
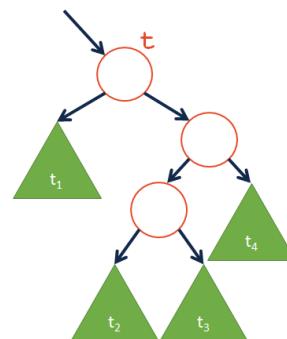
AVL Insertion

Pseudocode:
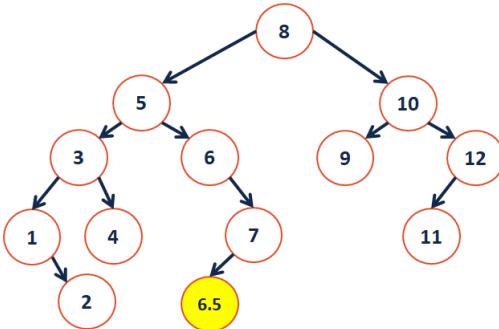
	AVL.h (snippet)
23	class TreeNode {
24	public:
25	T key;
26	unsigned height;
27	TreeNode *left;
28	TreeNode *right;
…	

AVL Insertion

	AVL.hpp
151	template <typename K, typename V>
152	void AVL<K, D>::_insert(const K & key, const V & data, TreeNode *& cur) {
153	if (cur == NULL) { cur = new TreeNode(key, data); }
157	else if (key < cur->key) { _insert(key, data, cur->left); }
160	else if (key > cur->key) { _insert(key, data, cur->right);}
166	_ensureBalance(cur);
167	}

119	template <typename K, typename V>
120	void AVL<K, D>::_ensureBalance(TreeNode *& cur) {
121	// Calculate the balance factor:
122	int balance = height(cur->right) - height(cur->left);
123	
124	// Check if the node is current not in balance:
125	if (balance == -2) {
126	int l_balance =
	height(cur->left->right) - height(cur->left->left);
127	if (l_balance == -1) { _____; }
128	else { _____; }
129	} else if (balance == 2) {
130	int r_balance =
	height(cur->right->right) - height(cur->right->left);
131	if(r_balance == 1) { _____; }
132	else { _____; }
133	}
134	
135	_updateHeight(cur);
136	};

AVL Insertion
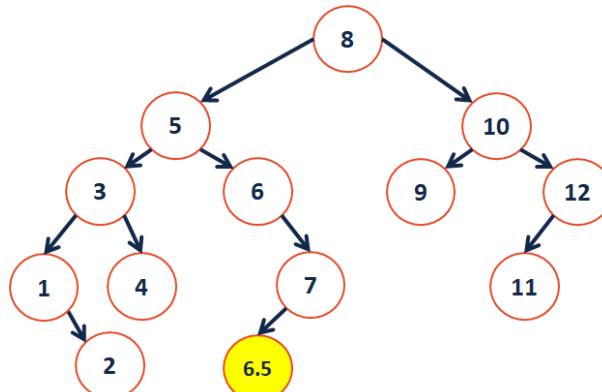
AVL Removal
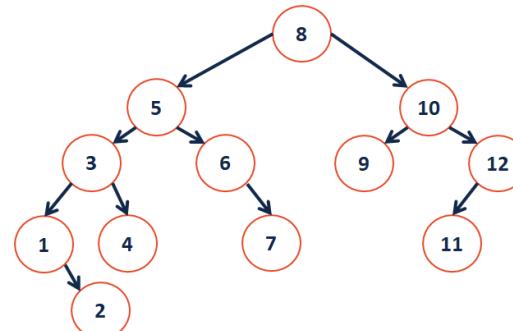
CS 225 – Things To Be Doing:
1. Theory Exam 2 starts tomorrow!
2. MP4 is released! EC deadline (Part 1) is this coming Monday!
3. lab_huffman in labs this week
4. Daily POTDs