

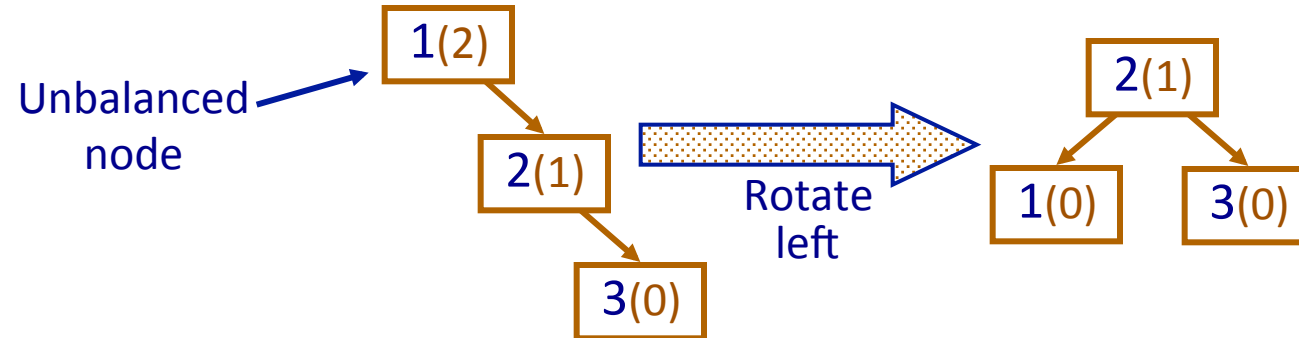
AVL Trees II

Concepts and Examples

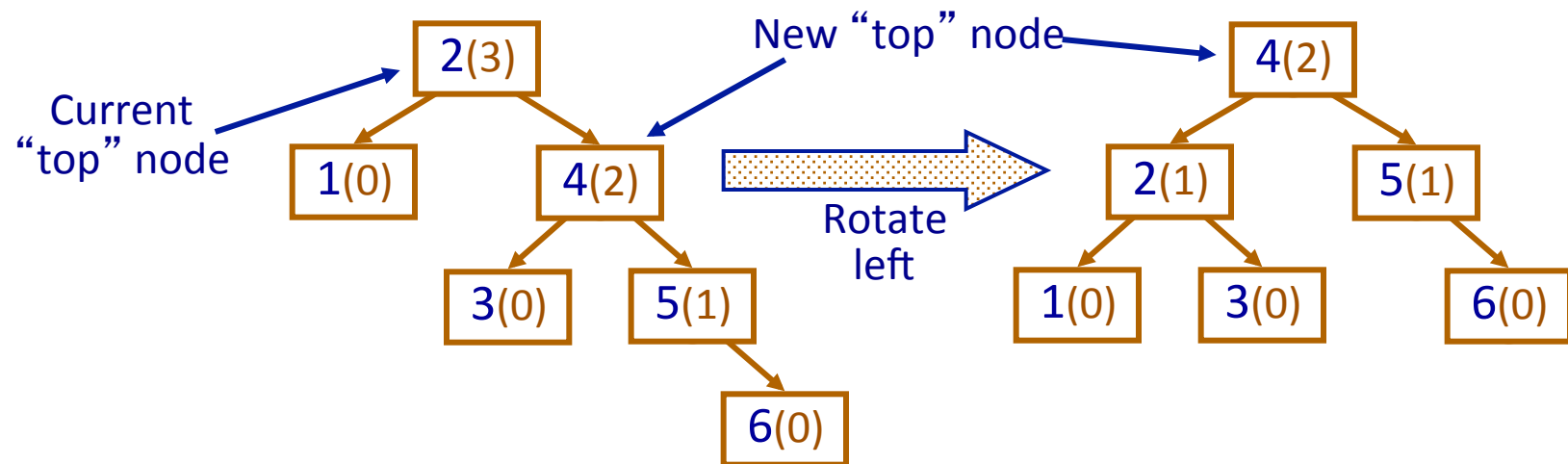
Goals

- Introduce AVL Tree Concepts
 - Rotation
 - Balancing
- Work through AVL Tree examples

Case 1: Single Rotation



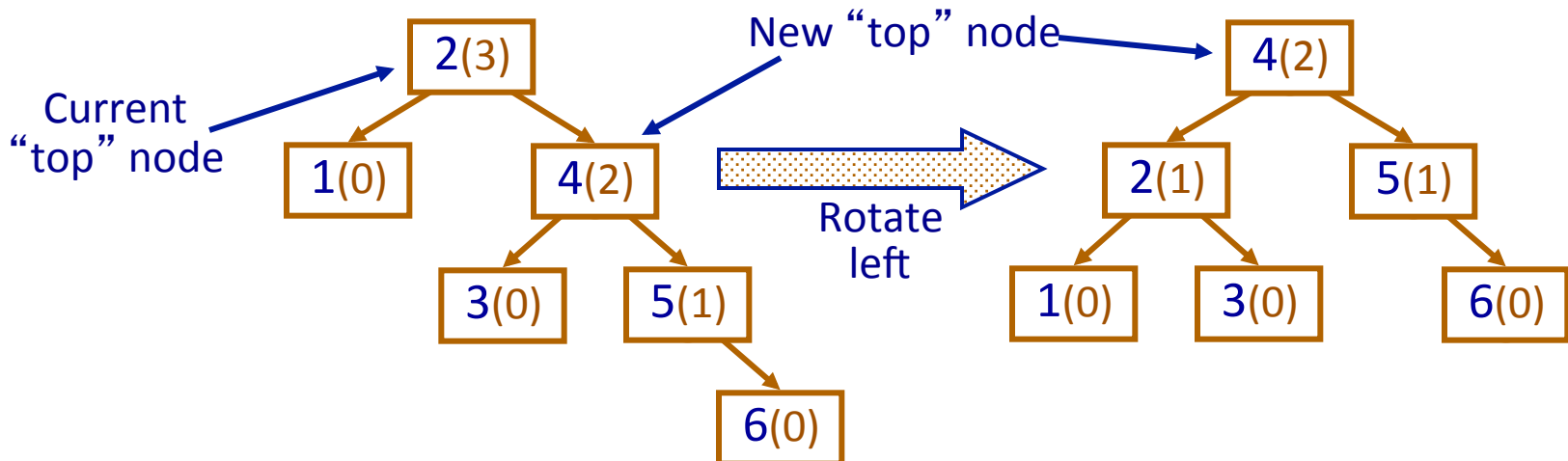
Case 1: Single Rotation Example 2



AVL Trees: Rotation Pseudocode

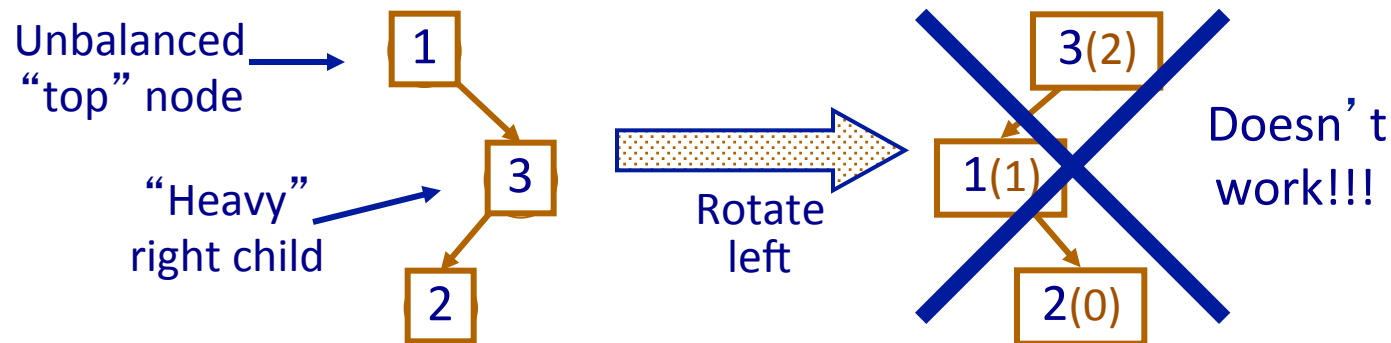
Pseudocode to rotate current (“top”) node *left*:

1. New top node is the current node’s right child
2. Current node’s new right child is the new top node’s (old right child’s) left child
3. New top’s left child is the current node
4. Set height of current node
5. Set height of new top node
6. Return new top node



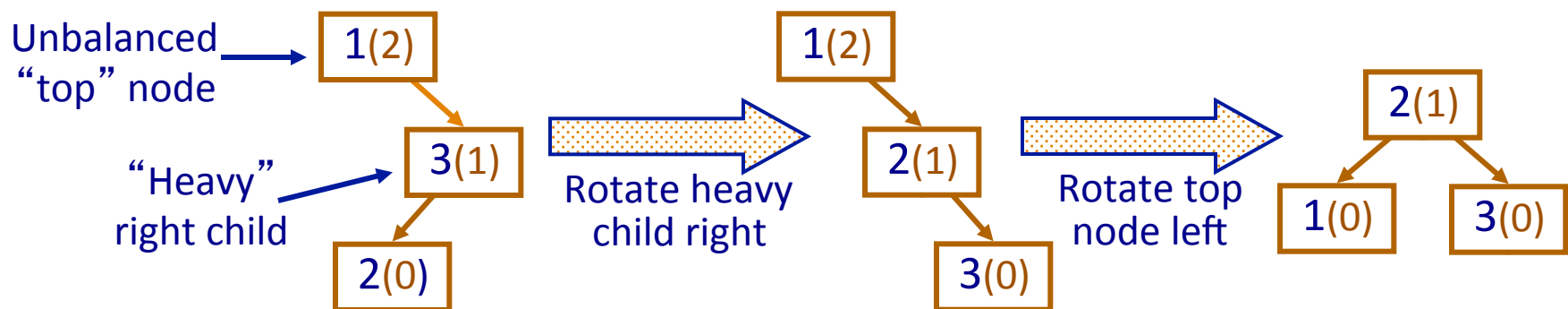
Case 2: Double Rotation

- Sometimes a single rotation will not fix the problem:
 - Can happen when an insertion is made on the left (or right) side of a node that is itself a heavy right (or left) child



Case 2: Double Rotation

- Fortunately, this case is easily handled by *rotating the child* before the regular rotation:
 - First rotate the heavy right (or left) child to the right (or left)
 - Rotate the “top” node to the left (or right)



AVL Trees: Balancing Pseudocode

Balancing pseudocode (to rebalance an unbalanced node):

If left child is tallest (by more than 1):

If left child is heavy on the right side: // Double rotation needed.

Rotate the left child to the left

Rotate unbalanced (“top”) node to the right

Else if right child is tallest (by more than 1)

If right child is heavy on the left side: // Double rotation needed.

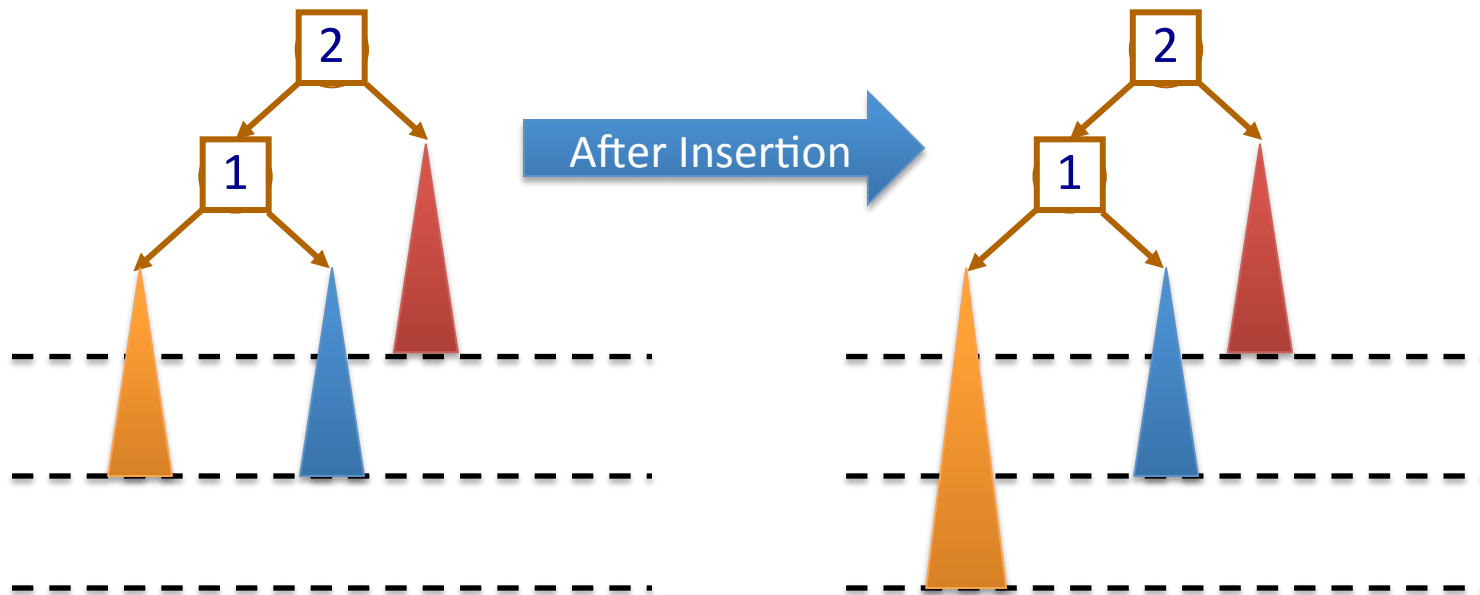
Rotate the right child to the right

Rotate unbalanced (“top”) node to the left

Return new “top” node

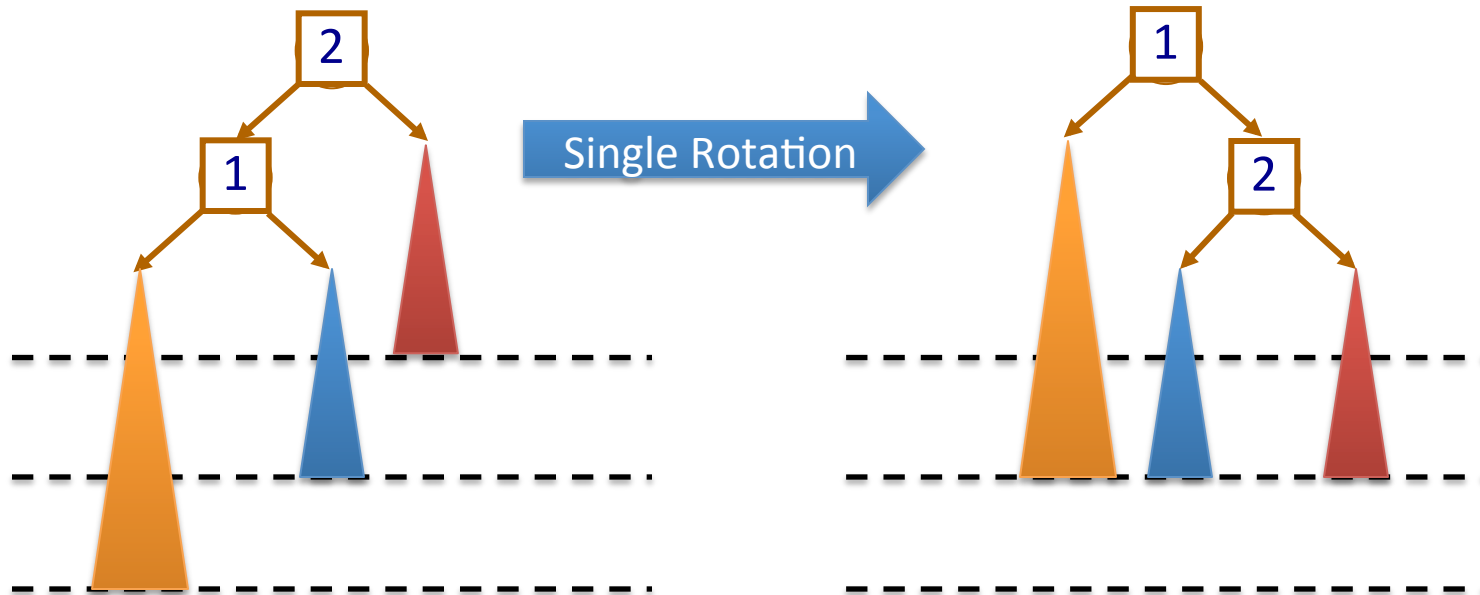
Case 1: Single Rotation

- Have a node (2) that is heavy on the left (1)
- Operation makes (2) unbalanced and that heavy child heavy on left



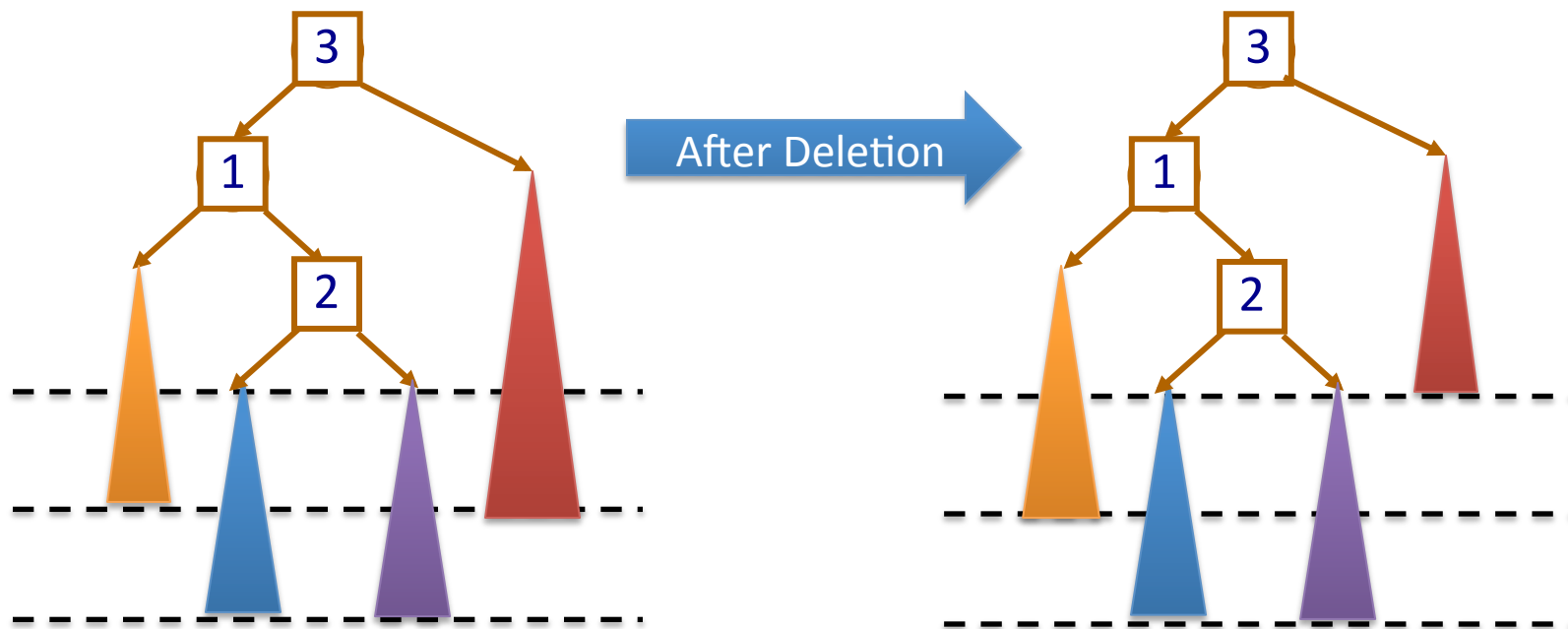
Case 1: Single Rotation

- Single rotation fixes the problem



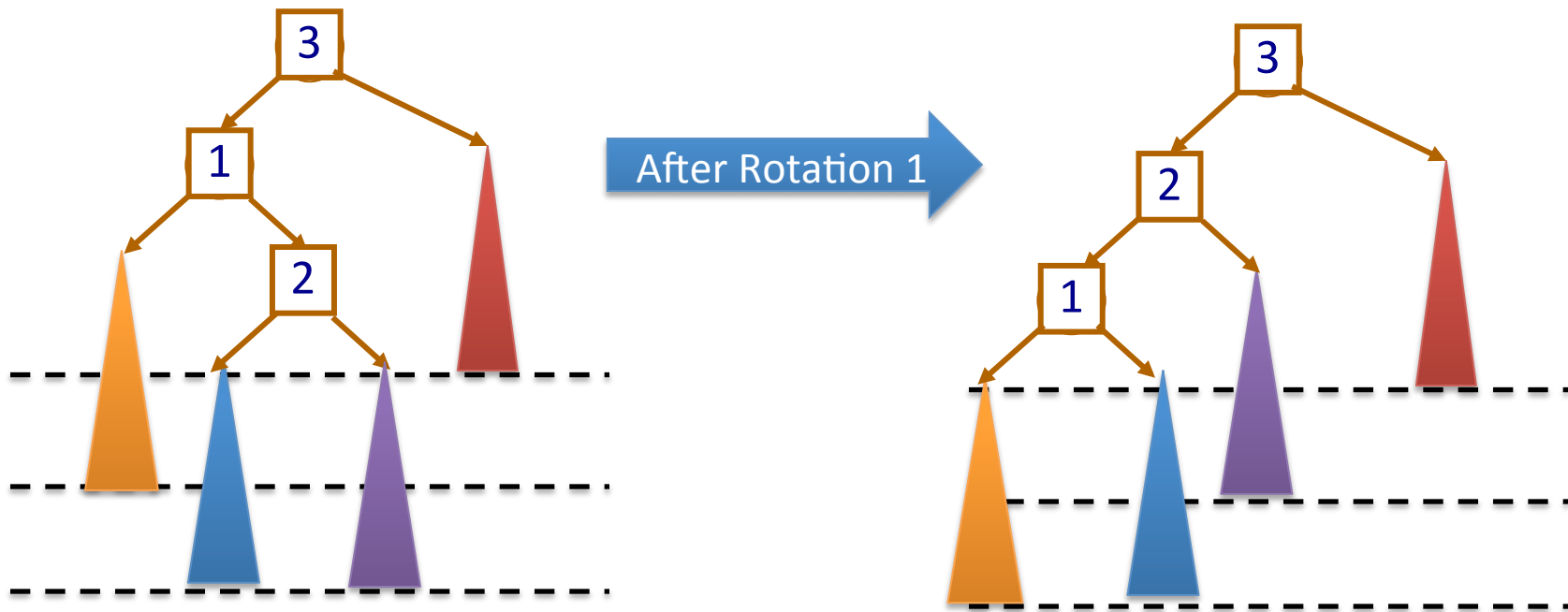
Case 2: Double Rotation

- Have a node (3) with a heavy left child (1)
- Operation makes (3) unbalanced and (1) is heavy on the right



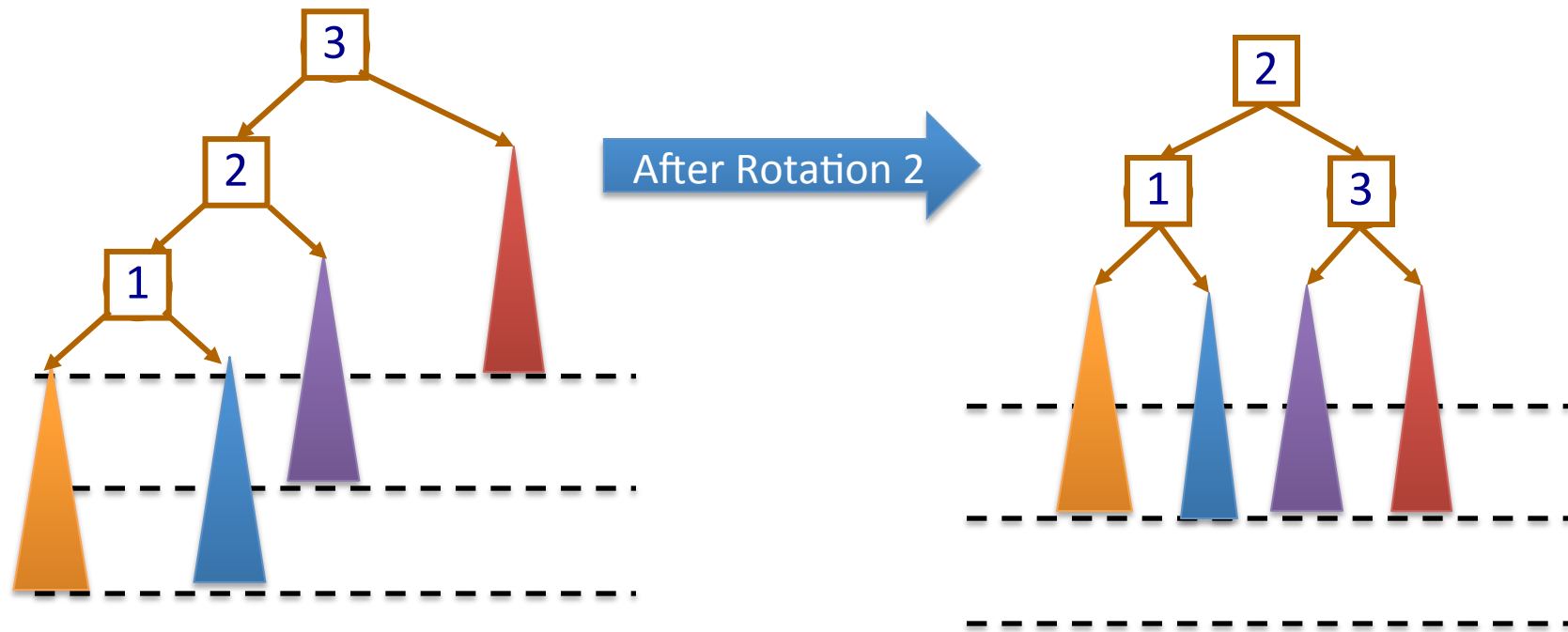
Case 2: Double Rotation

- First, rotate heavy child (1) to the left



Case 2: Double Rotation

- Next, rotate unbalanced node (3) to the right

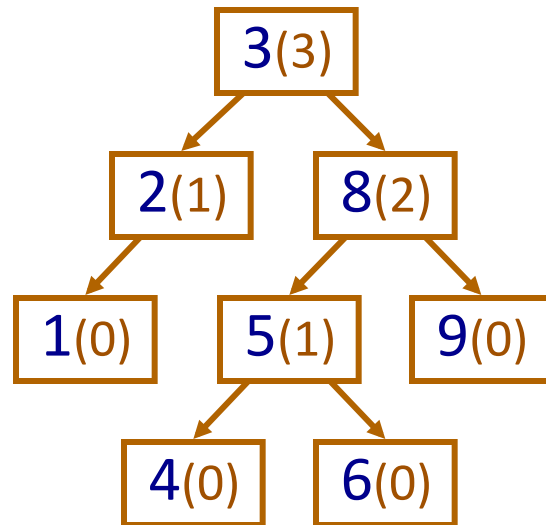


Rotations

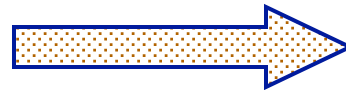
- Can be the result of additions or removals
- All cases hold for the “right” side as well, just replace all lefts with rights and all rights with lefts

AVL Trees: Example

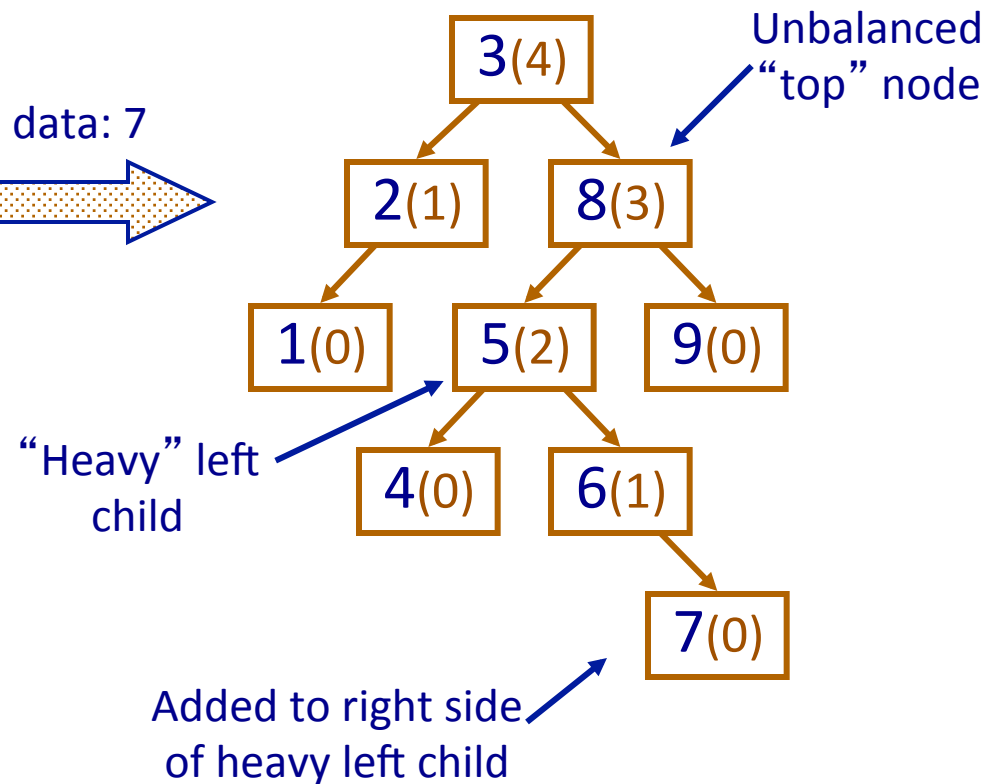
Balanced Tree



Add data: 7

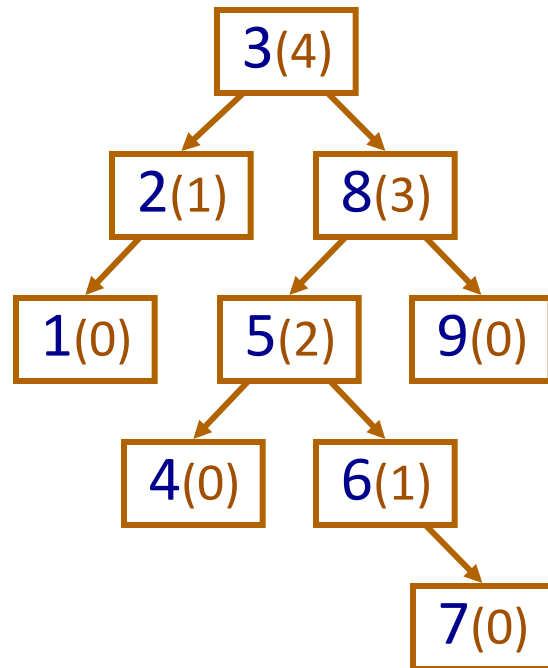


Unbalanced Tree

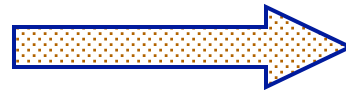


AVL Trees: Double Rotation Example

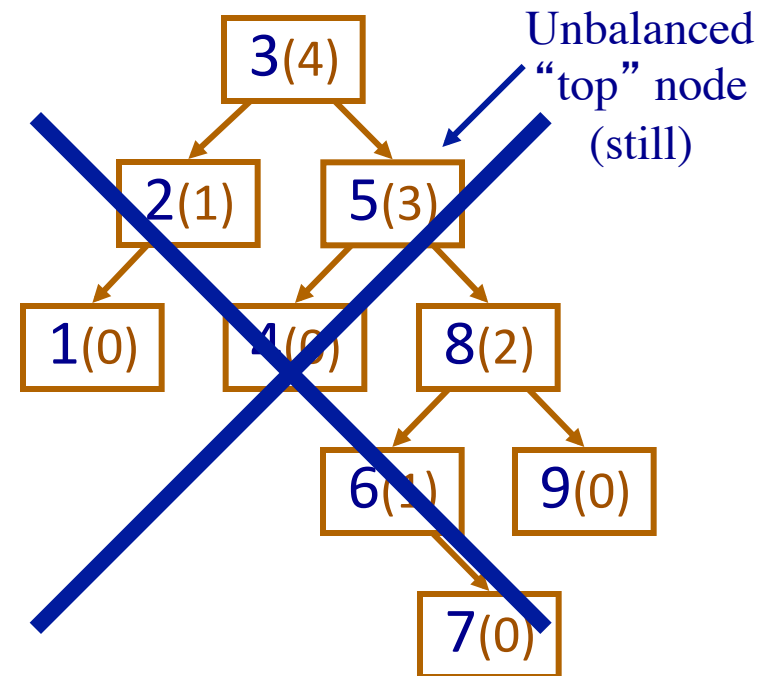
Unbalanced Tree



Single rotation

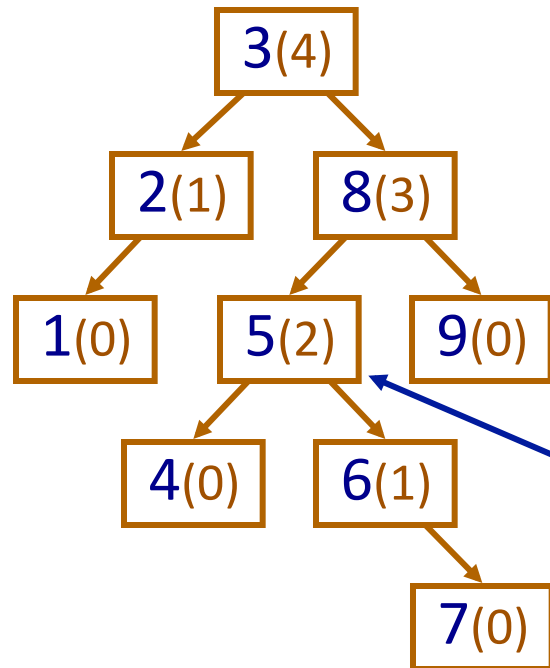


Tree Still Unbalanced

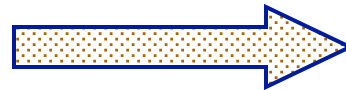


AVL Trees: Double Rotation Example

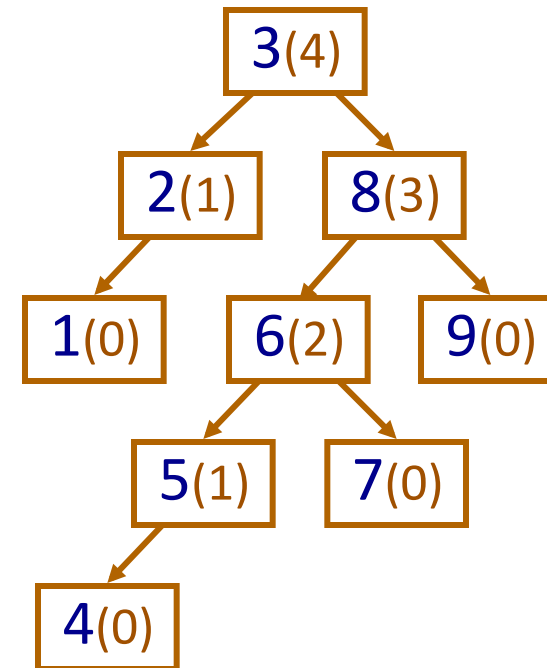
Unbalanced Tree



Rotate heavy child

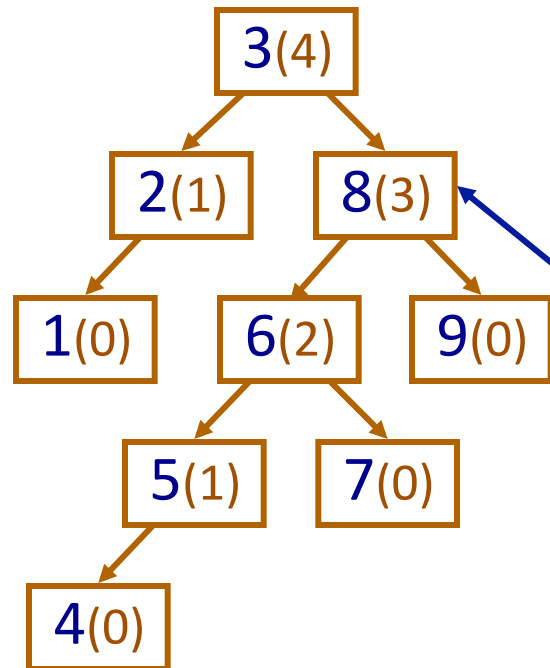


Tree Still Unbalanced, but ...

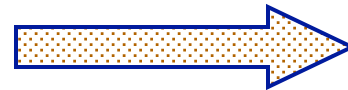


AVL Trees: Double Rotation Example

Unbalanced Tree
(after 1st rotation)

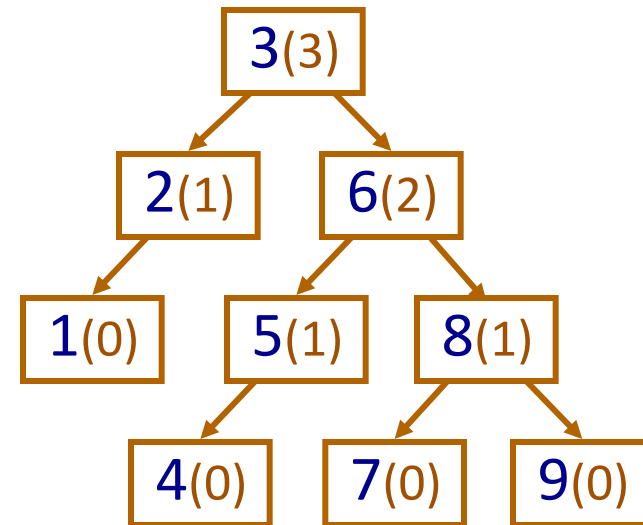


Rotate top node



Unbalanced
"top" node

Tree Now Balanced



- Worksheet: AVL Practice