AVL Trees: Deletion

Computer Science Department
University of Central Florida

COP 3502 – Computer Science I

Insertion Revisited

- AVL Trees: Insertion
  - Let's take another look at insertion into AVL Trees
  - Hopefully this will be a bit easier than previous slides
  - Assuming you only have two nodes in your tree,
  - what are the two possible trees you may have?

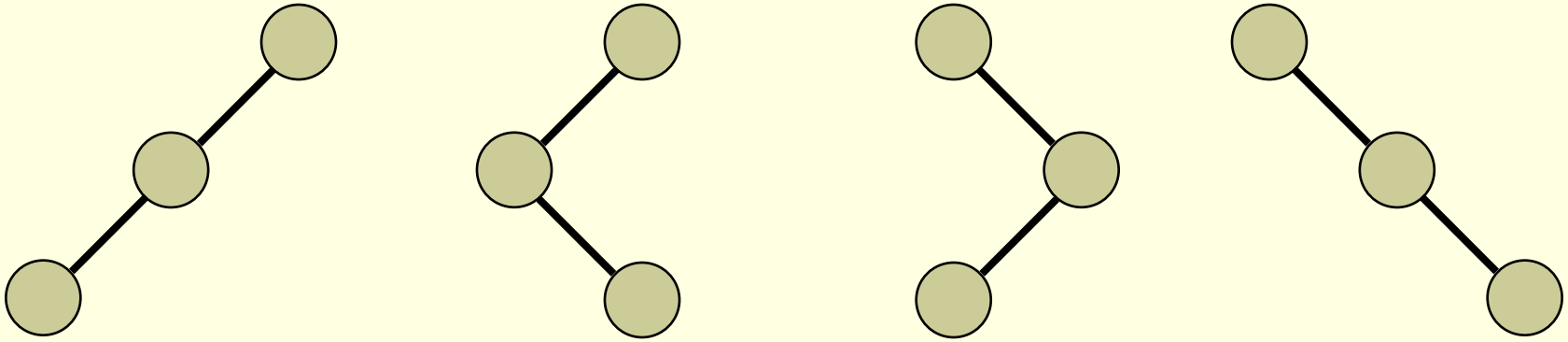
Insertion Revisited

- AVL Trees: Insertion
  - Given these two trees, if we want to create an imbalance, where must we insert?

    - Clearly, we must insert at the lower of the 2 nodes
    - This will create a scenario where the left subtree has a height that is 2 greater than the right subtree
      - Or the opposite for the other tree depicted
  - Now, from these two trees, draw all FOUR possible trees that can be created by inserting a new node

Insertion Revisited

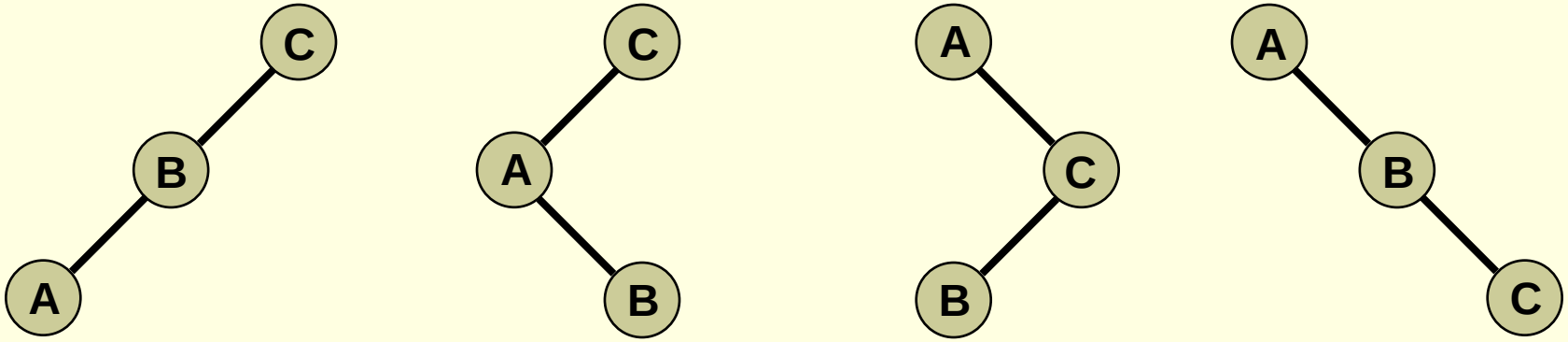
- AVL Trees: Insertion
  - Here are all four unbalanced trees that we can make from three nodes:



    - Now, label these nodes with the labels A, B, and C
      - Where A is the smallest of the three nodes, B is the middle node, and C is the largest.
      - The inorder traversal of each tree should be A, B, C

Insertion Revisited

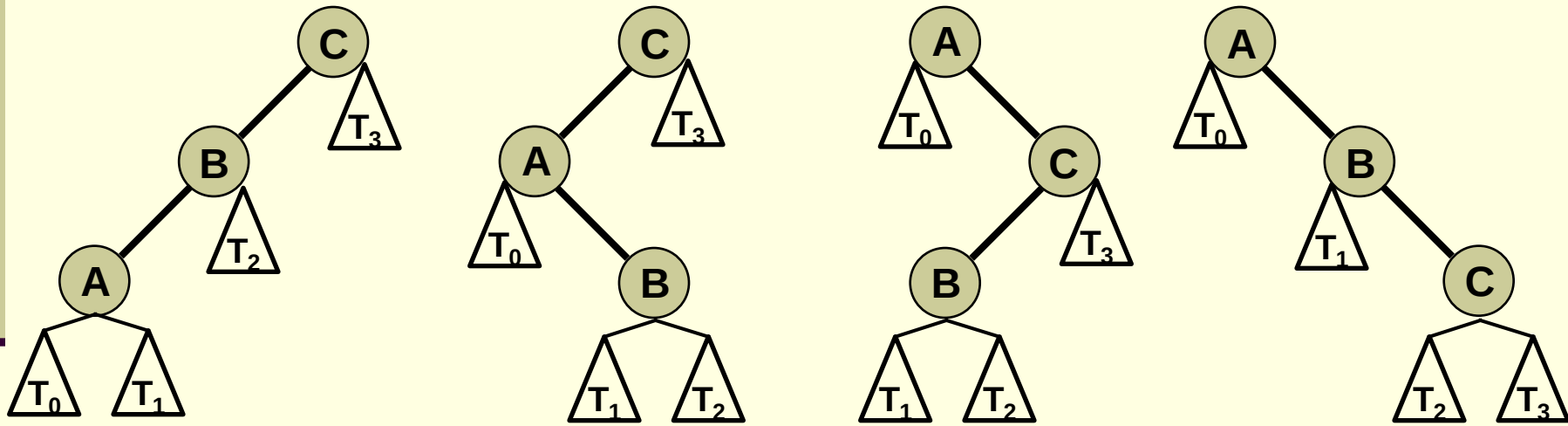
- AVL Trees: Insertion
  - Here are all four trees with the node lables in their inorder listing:



    - Any time an imbalance occurs, it is localized to three nodes and their four subtrees
      - These are the four possibilities
      - Now we add in the depiction of the four subtrees of A, B, and C

Insertion Revisited

- AVL Trees: Insertion
  - Here are all four trees with the node lables in their **inorder** listing with subtrees in their **inorder** listing:



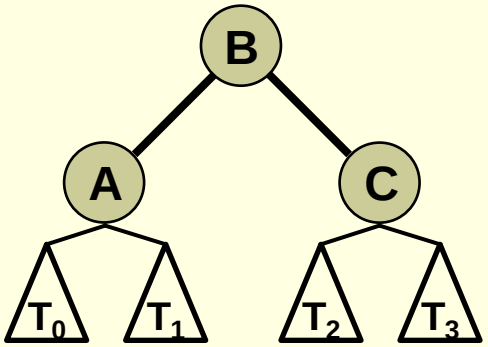
  - We denote the four subtrees as T_0, T_1, T_2, and T_3
  - And they are listed in their inorder listing

Insertion Revisited

- AVL Trees: Insertion
  - So what is the purpose of all this?
  - We said this method is supposedly MUCH easier than dealing with the various rotations of the tree
  - So we've done all this labeling
    - Finding nodes 'A', 'B', and 'C' and labeling them as such
  - How they heck does this help us???
  - Here ya go…

Insertion Revisited

- AVL Trees: Insertion
  - Part 1: Once an insertion causes an imbalance, find and label the nodes ‘A’, ‘B’, and ‘C’
  - Part 2: Once the nodes are labeled, no matter what structural imbalance occurred, they can all be fixed the same way:

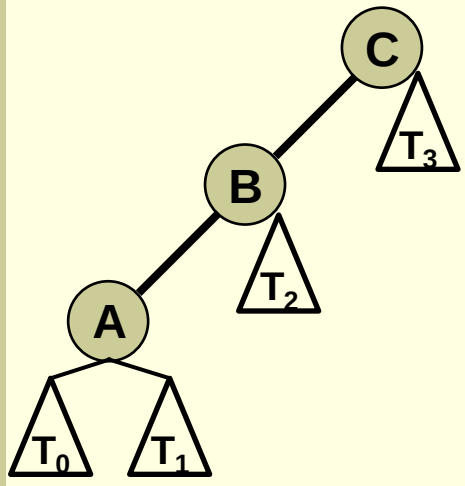


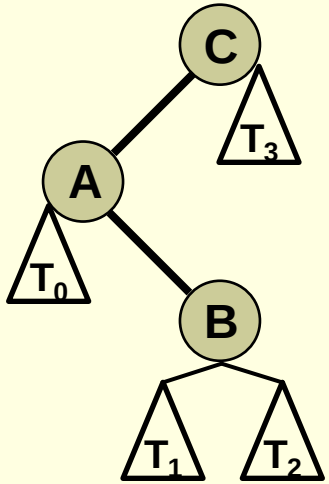
    - Simply restructure those three nodes, and their four respective subtrees, as shown above, and the imbalance will be corrected!

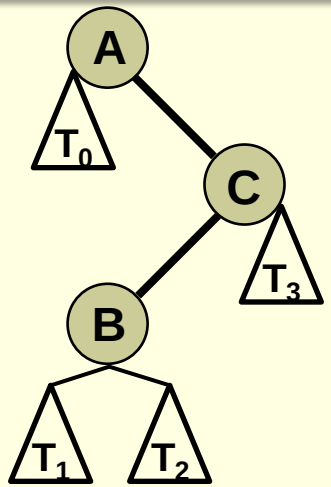
Insertion Revisited

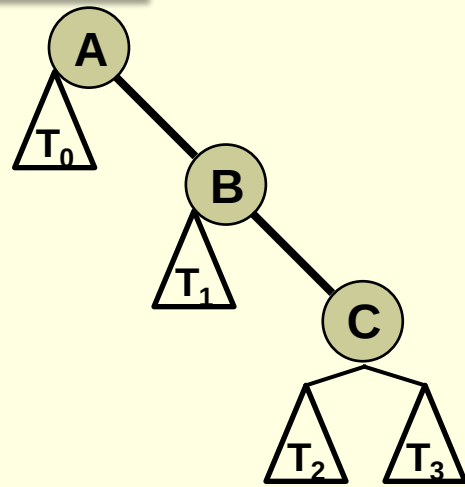
- AVL Trees: Insertion

All 4 of these trees:

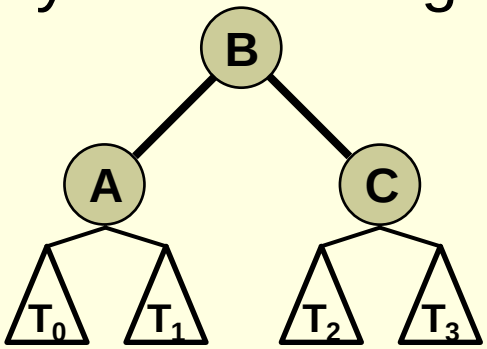








- Can be fixed by restructuring into this:

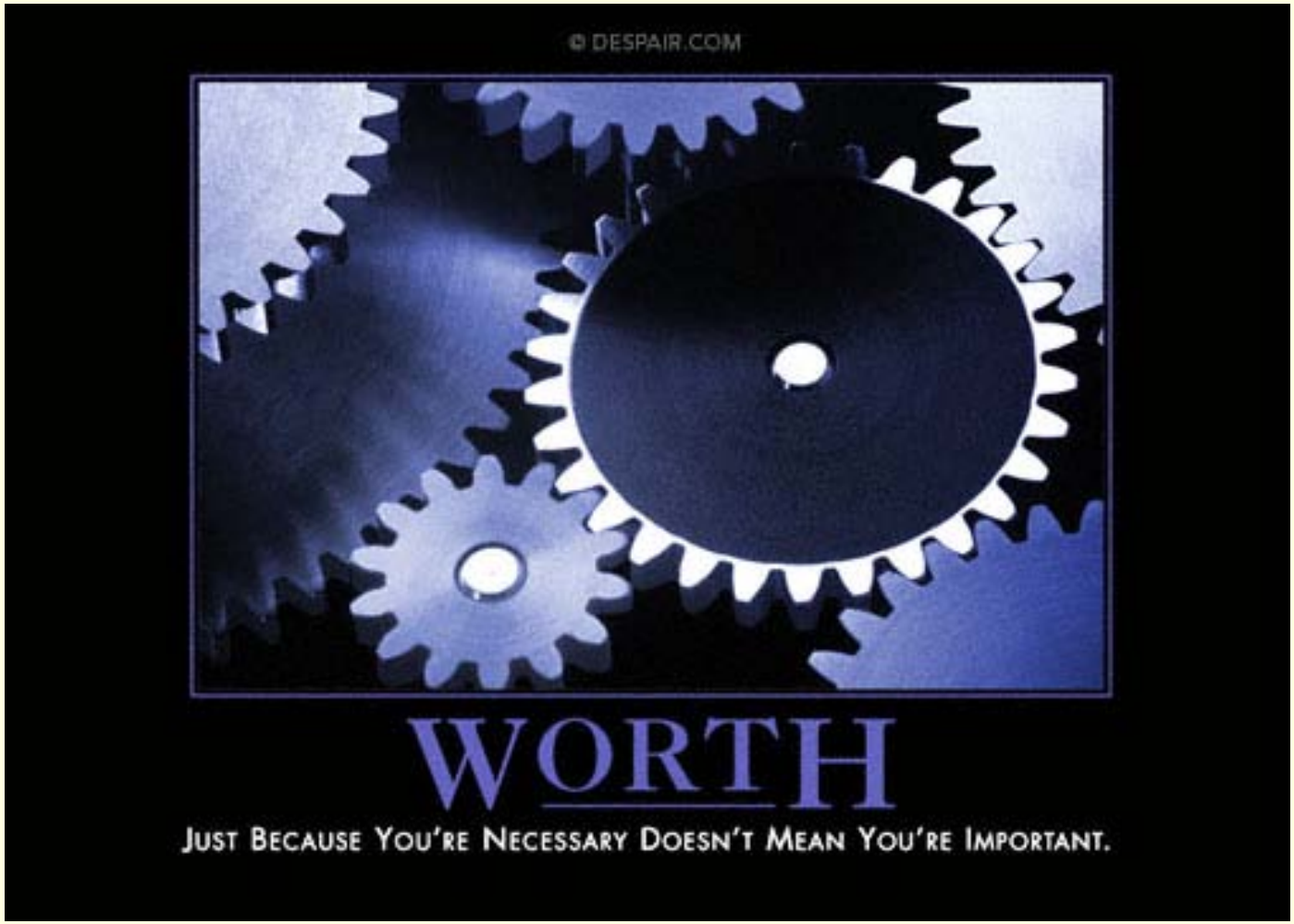


Brief Interlude: FAIL Picture

AVL Trees: Deletion

WASN'T THAT MOMENTOUS!

Daily Demotivator



AVL Trees: Deletion

Computer Science Department
University of Central Florida

COP 3502 – Computer Science I