AVL Trees: Deletion

Computer Science Department
University of Central Florida

COP 3502 – Computer Science I

AVL Trees: Deletion

- AVL Trees: Deletion
  - **Practical Rules**:
    1. Start by doing a normal BST deletion
    2. Then, begin updating balance factors of nodes along the path from the deletion point to the root
    3. As soon as you find the first node out of balance, mark that node as one of your three "restructuring nodes"

    - Thus far, this looks just like insertion
    - But here's where things change
      - a lot

AVL Trees: Deletion

- AVL Trees: Deletion
  - Remember: Insertion Rules
    1. Start finding the balance factors of ALL nodes along the path from the insertion point to the root
    2. As soon as you find the first node out of balance, mark that node as one of your three "restructuring nodes"
    3. Then, take two steps, back down, towards the insertion point and mark those two nodes as well.

    - Then we have rules 4 and 5
    - But rule 3 is where the change occurs
    - For insertion, we take two steps back down, towards the insertion point

AVL Trees: Deletion

- AVL Trees: Deletion
  - Difference with Deletion Rules
    - But for deletion, we do NOT take two steps down, towards the point of deletion
    - This would NOT make any sense!
    - Why?
    - When we insert, we are making that portion of the tree TALLER!
      - So if an imbalance occurs at some node, the TALLEST subtree, of said node, needs to be restructured
      - And which subtree is the taller one?
      - The one that has a path from the out-of-balance (OOB) node to the point of insertion!

AVL Trees: Deletion

- AVL Trees: Deletion
  - Difference with Deletion Rules
    - But for deletion, we do NOT take two steps down, towards the point of deletion
    - This would NOT make any sense!
    - Why?
    - When we delete, however, we are making that particular portion of the tree SHORTER!
      - So if an imbalance occurs at some node, as with insertion, the restructuring must occur on the TALLEST subtree of said node
      - And is that tallest subtree on the path from root to deletion point?
      - Absolutely NOT! Because that path just got SHORTER!!!
      - **So the GENERAL rule**: **<u>restructure down TALLEST path</u>**

AVL Trees: Deletion

- AVL Trees: Deletion
  - **Practical Rules**:
    1. Start by doing a normal BST deletion
    2. Then, begin updating balance factors of nodes along the path from the deletion point to the root
    3. As soon as you find the first node out of balance, mark that node as one of your three "restructuring nodes"
    4. Then, take a step down the TALLER subtree
    5. Then, again, take a step down the TALLER subtree
       - So that was 2 steps: now **mark** both of those nodes as well
    6. Label those 'A, B, C' nodes appropriately(and subtrees)
    7. Restructure those three nodes (and their subtrees)
    8. After restructuring, recursively check BFs up to root

AVL Trees: Deletion

- AVL Trees: Deletion
  - Notes:
    - with an insertion, at MOST one node needs to be rebalanced
    - but for deletion, there may be MULTIPLE nodes that need to be rebalanced
    - At any point during the previous, restructuring algorithm, ONLY one node will ever be unbalanced
    - However, what may happen is when that node is fixed, an "error" may propagate to an ancestor node.
    - So once we fix the OOB node via restructuring, we need to make sure to keep checking the BFs on the path, to the root, and constantly restructure as needed

AVL Trees: Deletion

- AVL Trees: Deletion
  - Choosing ‘A, B, C’ Nodes:
    - Remember: the nodes A, B, and C are always on the TALLEST path to the bottom of the tree
      - So when we find an imbalanced node after deleting, the node to the opposite side is guaranteed to be down the longer path
    - So that would give us two of the three nodes we need
    - From there, we have a choice for the third node of A, B, C

    1) If one side is longer than the other, choose that side
    2) If the two sides are equal, you cannot just choose the “taller” side
       - So follow this rule: **Go to the same side as the parent is to the grandparent**

examples will clarify…

Brief Interlude: FAIL Picture

Daily UCF Bike FAIL

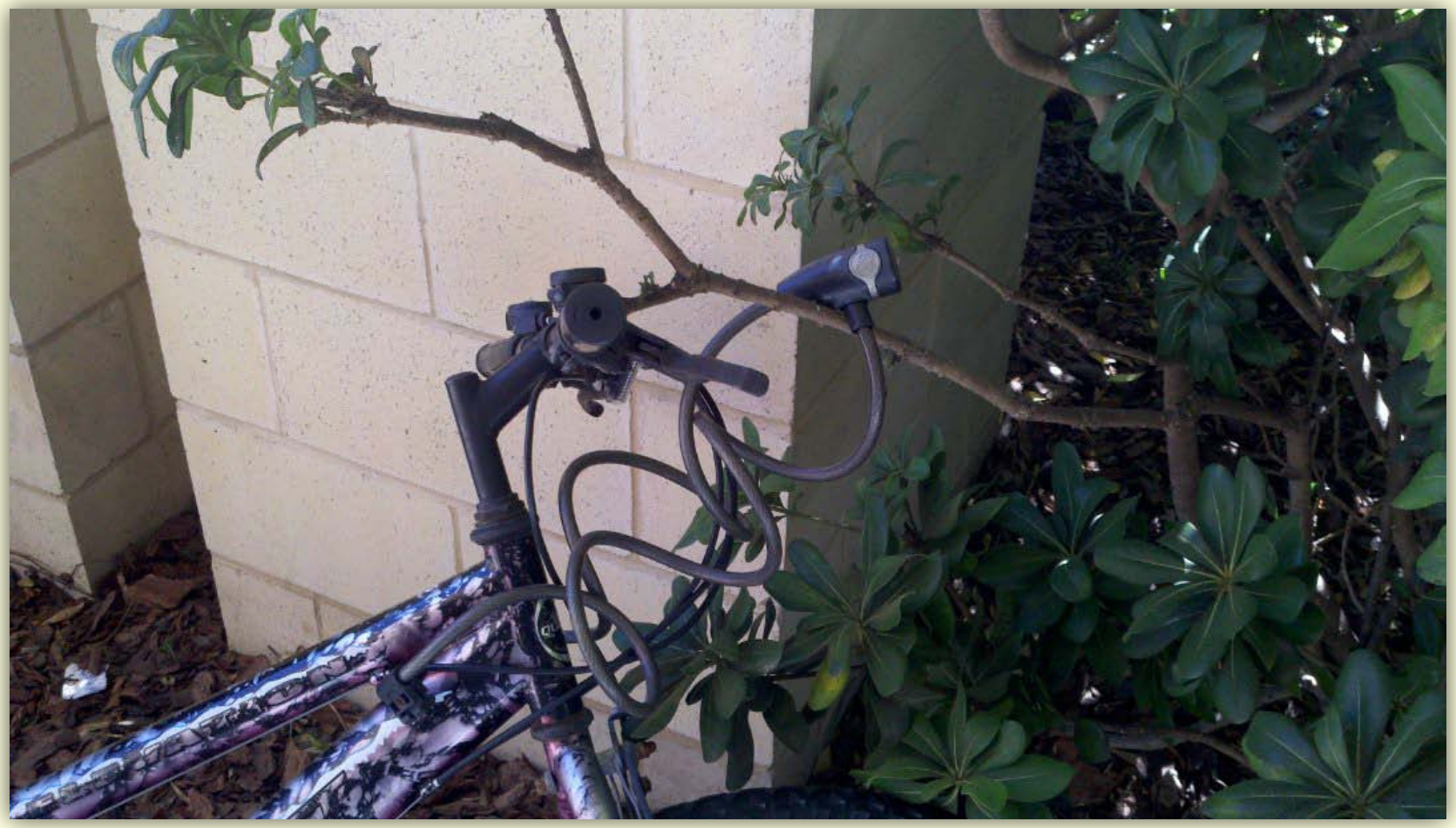
- Thoughts:
  1. Brace Yourself!
  2. This was from last semester
     - In front of HPA
  3. The student SWORE that he did not set it up
  4. Just saw it, was literally dumbfounded, literally lost brain cells just by looking at it, and finally, thankfully, took a couple photos.

- **<u>Final WARNING</u>**:
  - Viewing will almost assuredly reduce brain function

Daily UCF Bike FAIL

Daily UCF Bike FAIL

Daily UCF Bike FAIL



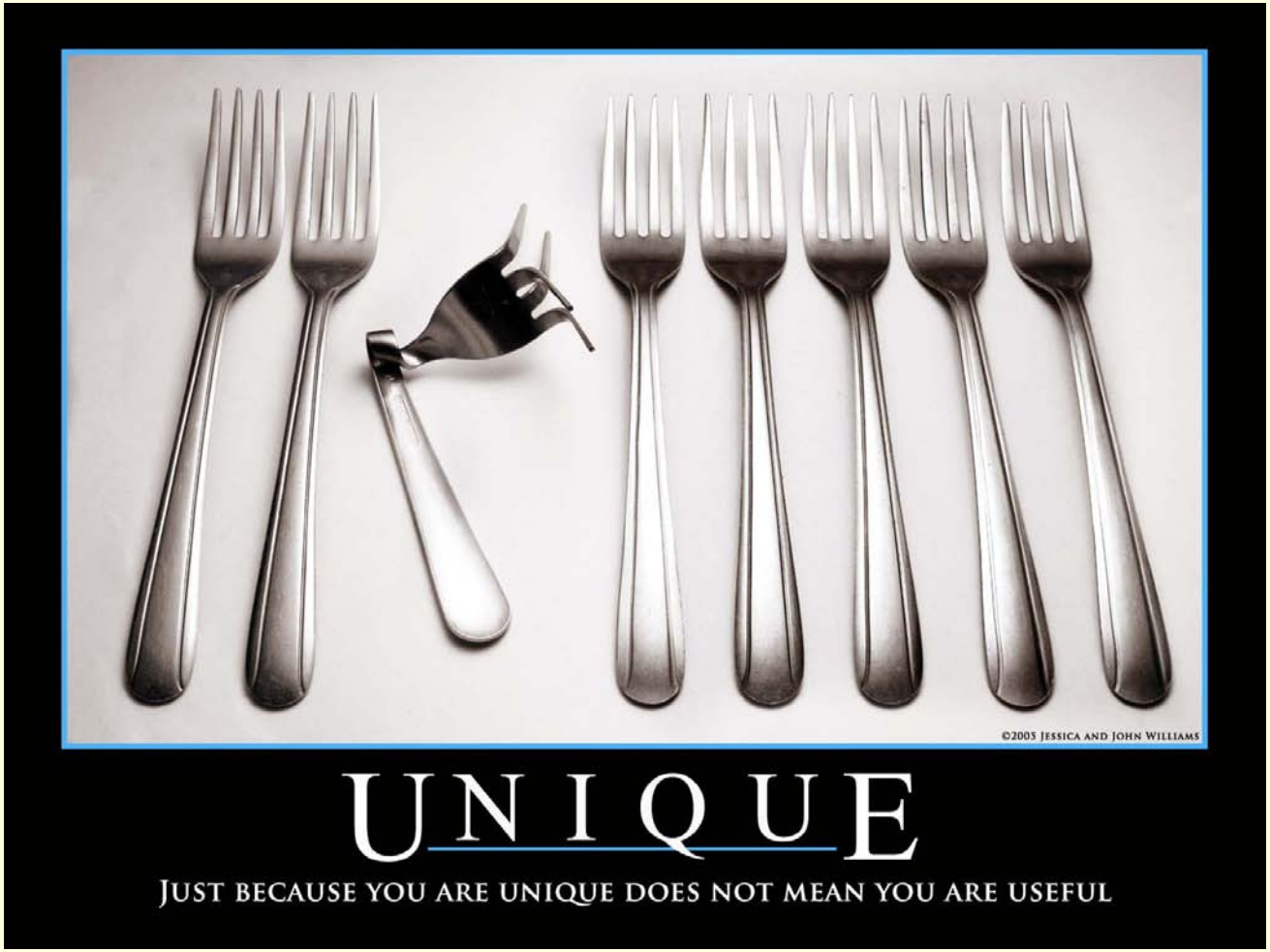
AVL Trees: Deletion

- More examples given in class!
- See PDF of Arup's Deletion notes
  - maybe not the most exciting notes
  - but it has the same examples

AVL Trees: Deletion

WASN'T THAT ASTOUNDING!

Daily Demotivator



AVL Trees: Deletion

Computer Science Department
University of Central Florida

COP 3502 – Computer Science I