CSSE 230 Day 15

AVL insert/Delete
AVLTree practice
Doublets Work time

Agenda

- Answers to your questions.
- By Wednesday at noon:
  - EditorTrees partner preference survey
  - Hardy "resubmission" (if you wish)
    - Grab screenshot from Eclipse
    - Submit to dropbox on Moodle
- AVL insertion/deletion details and practice
- Work time

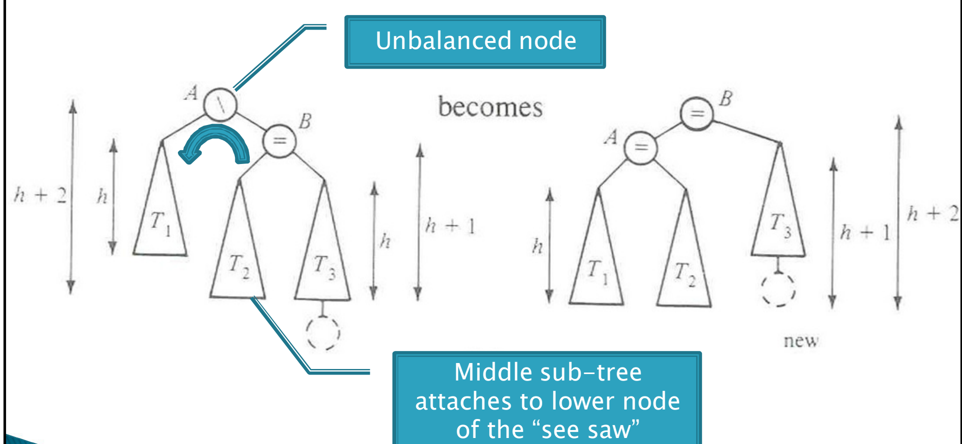
- A word on in-class work time" in general!
  - You should not leave or work on other courses unless you have finished the next programming assignment and next written assignment.
  - The idea is to work on things while you can get help if questions come up.

Overview of "rebalance after insertion"

- p = parent of inserted node
- while p != null
  - if p.balanceCode is '='
    - set code to '/' or '\' as appropriate
    - p = p.getParent()
  - else if p.balanceCode indicates "insertion was in shorter subtree"
    - change code to '='
    - break
  - else //insertion was into taller side.
    - do the appropriate rotation
    - break

Single Left Rotation

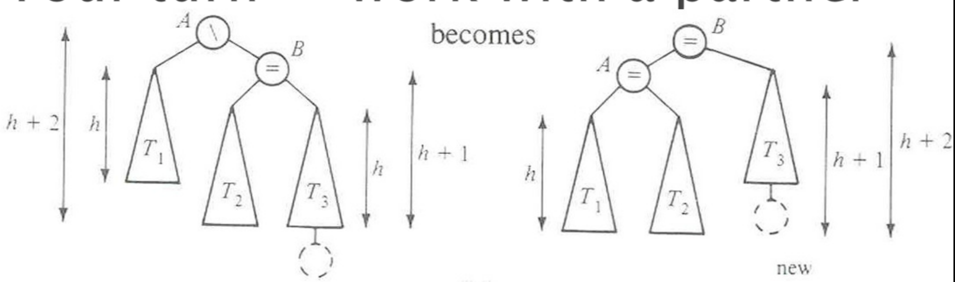
Q9-10



Diagrams are from *Data Structures* by E.M. Reingold and W.J. Hansen

Q15 (11-14 later)

Your turn — work with a partner

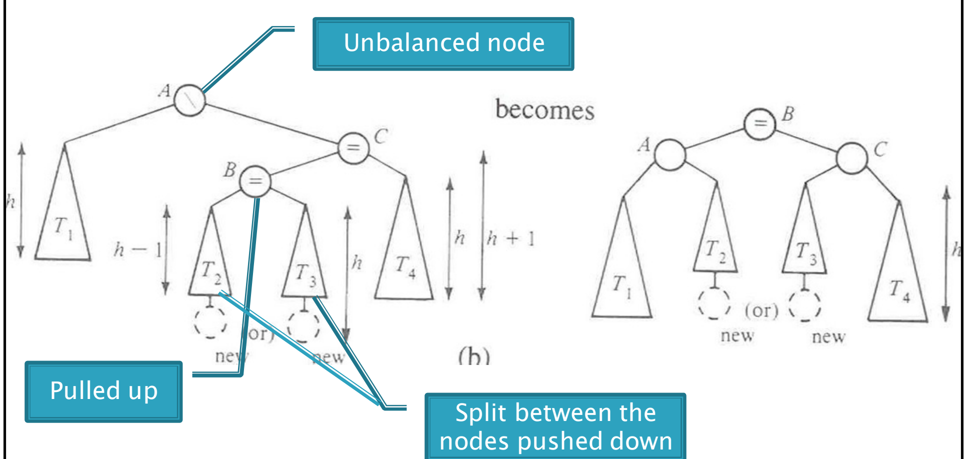


- Write the method:
-
```
static BalancedBinaryNode singleRotateLeft (
     BalancedBinaryNode parent,    /* A */
     BalancedBinaryNode child      /* B */   ) {

}
```
- Returns a reference to the new root of this subtree.
- Fields to change are *leftChild*, *rightChild*, *balanceCode*
- Possible values for balanceCode: / = \

Q11-12

Double Left Rotation



Weiss calls this “right-left double rotation”

More practice— (sometime after class)

- Write the method:
-
```
BalancedBinaryNode doubleRotateRight (
  BalancedBinaryNode parent,      /* A */
  BalancedBinaryNode child,       /* C */
  BalancedBinaryNode grandChild   /* B */ ) {


}
```
- Returns a reference to the new root of this subtree.
- Rotation is mirror image of double rotation from an earlier slide

Q13-14, 16-17   Submit Quiz 13

O(log N)?

- Both kinds of rotation leave height the same as before the insertion!
- Is insertion plus rotation cost really O(log N)?

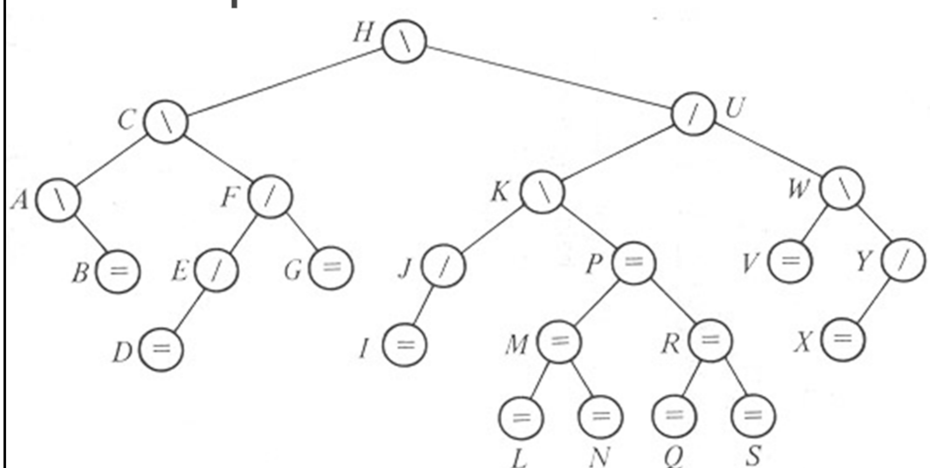
Insertion/deletion in AVL Tree:	O(log n)
Find the imbalance point (if any):	O(log n)
Single or double rotation: *in deletion case, may have to do O(log N) rotations*	O(1)
Total work:	O(log n)

Which kind of rotation to do after an insertion?

Depends on the first two links in the path from the lowest node that has the imbalance (A) down to the newly-inserted node.

First link (down from A)	**Second link** (down from A's child)	**Rotation type** (rotate "around A's position")
Left	Left	Single right
Left	Right	Double right
Right	Right	Single left
Right	Left	Double left

A sample AVL tree



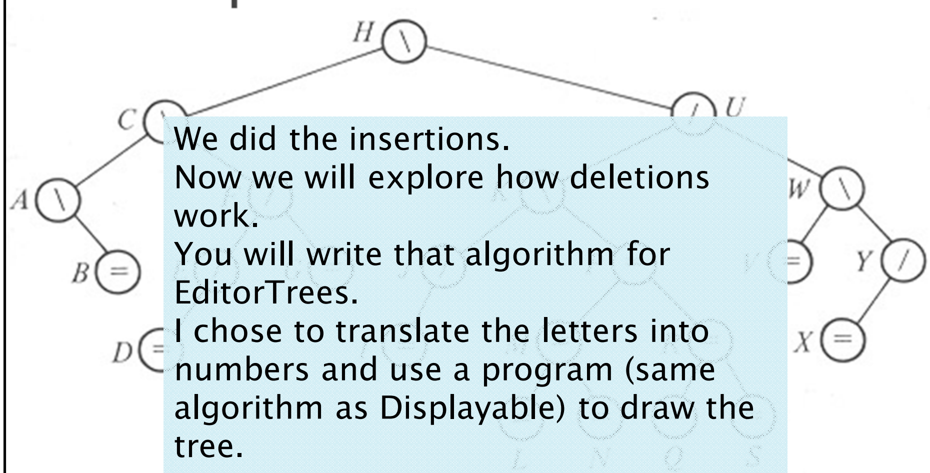
Insert **HA** into the tree, then **DA**, then **O**.
Delete **G** from the original tree, then **I**, **J**, **V**.

Recap: Which kind of rotation to do after an insertion?

Depends on the first two links in the path from the lowest node that has the imbalance (A) down to the newly-inserted node.

First link (down from A)	**Second link** (down from A's child)	**Rotation type** (rotate "around A's position")
Left	Left	Single right
Left	Right	Double right
Right	Right	Single left
Right	Left	Double left

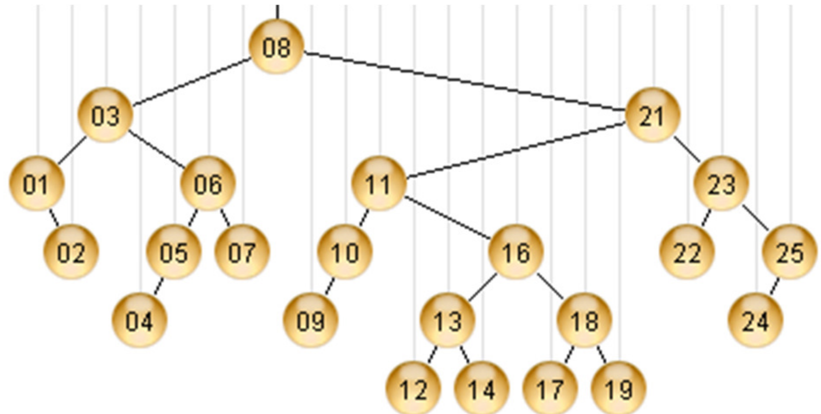
A sample AVL tree



We did the insertions.
Now we will explore how deletions work.
You will write that algorithm for EditorTrees.
I chose to translate the letters into numbers and use a program (same algorithm as Displayable) to draw the tree.

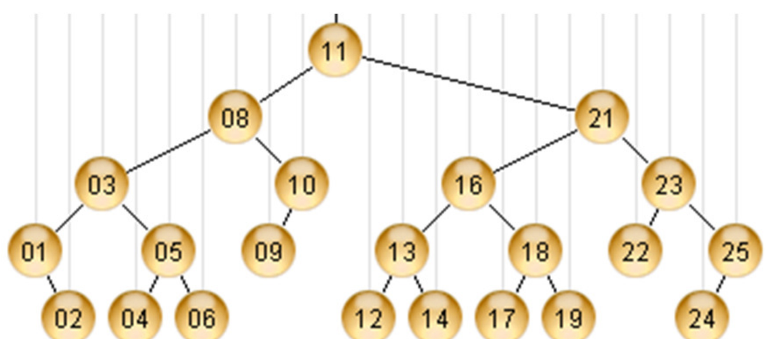
Insert **HA** into the tree, then **DA**, then **O**.
Delete **G** from the original tree, then **I**, **J**, **V**.

Delete 7 and Rebalance

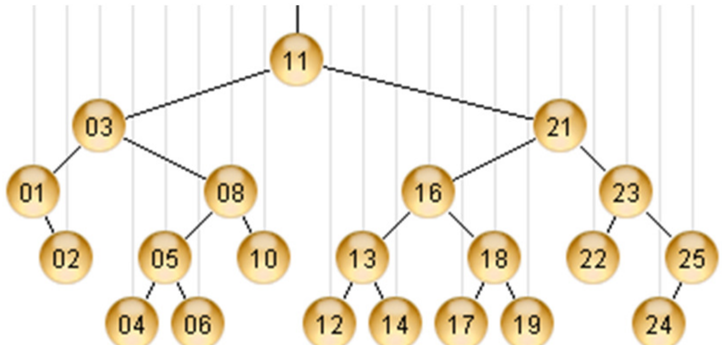


http://webdiis.unizar.es/asignaturas/EDA/AVLTree/avltree.html

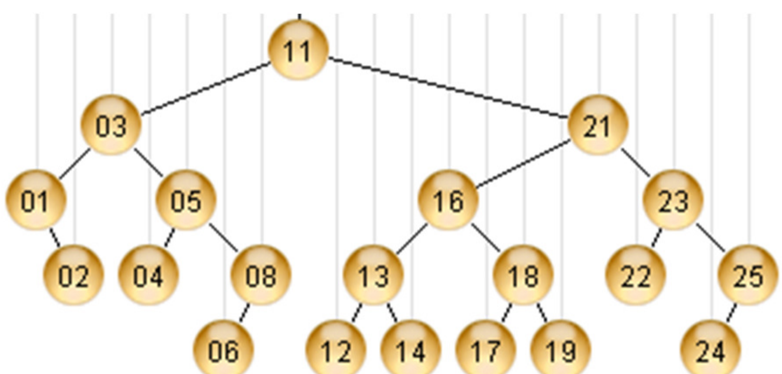
Delete 9 and Rebalance



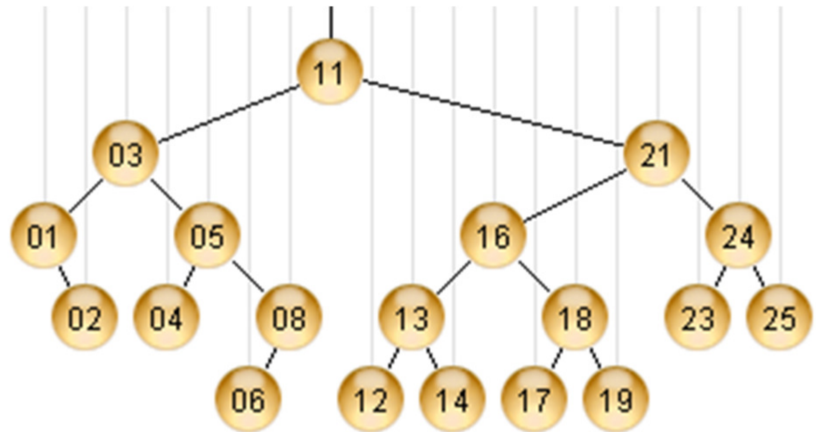
Delete 10 and Rebalance



Delete 22 and Rebalance



Final result



1-4

Your turn (with a partner or two)

- Start with an empty AVL tree.
- Add elements in the following order; do the appropriate rotations when needed.
  - 1 2 3 4 5 6 11 13 12 10 9 8 7
- How should we rebalance if each of the following sequences is deleted from the final tree above?
  - (10 9 7 8) (13) (1 5)
  - For each of the three sequences, start with the original 13-element tree. E.g. when deleting 13, assume 10 9 8 7 are still in the tree.

Work with your Doublets partner.
When you finish, work on Doublets or Threaded.
Or write the rotateDoubleRight code from a yesterday's slide.