CSSE 230 Day 15

AVL insert/Delete
AVLTree practice
Doublets Work time

Agenda

- Answers to your questions.
- By tonight:
  - EditorTrees partner preference survey

- AVL insertion/deletion practice
- Work time for doublets

- 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.
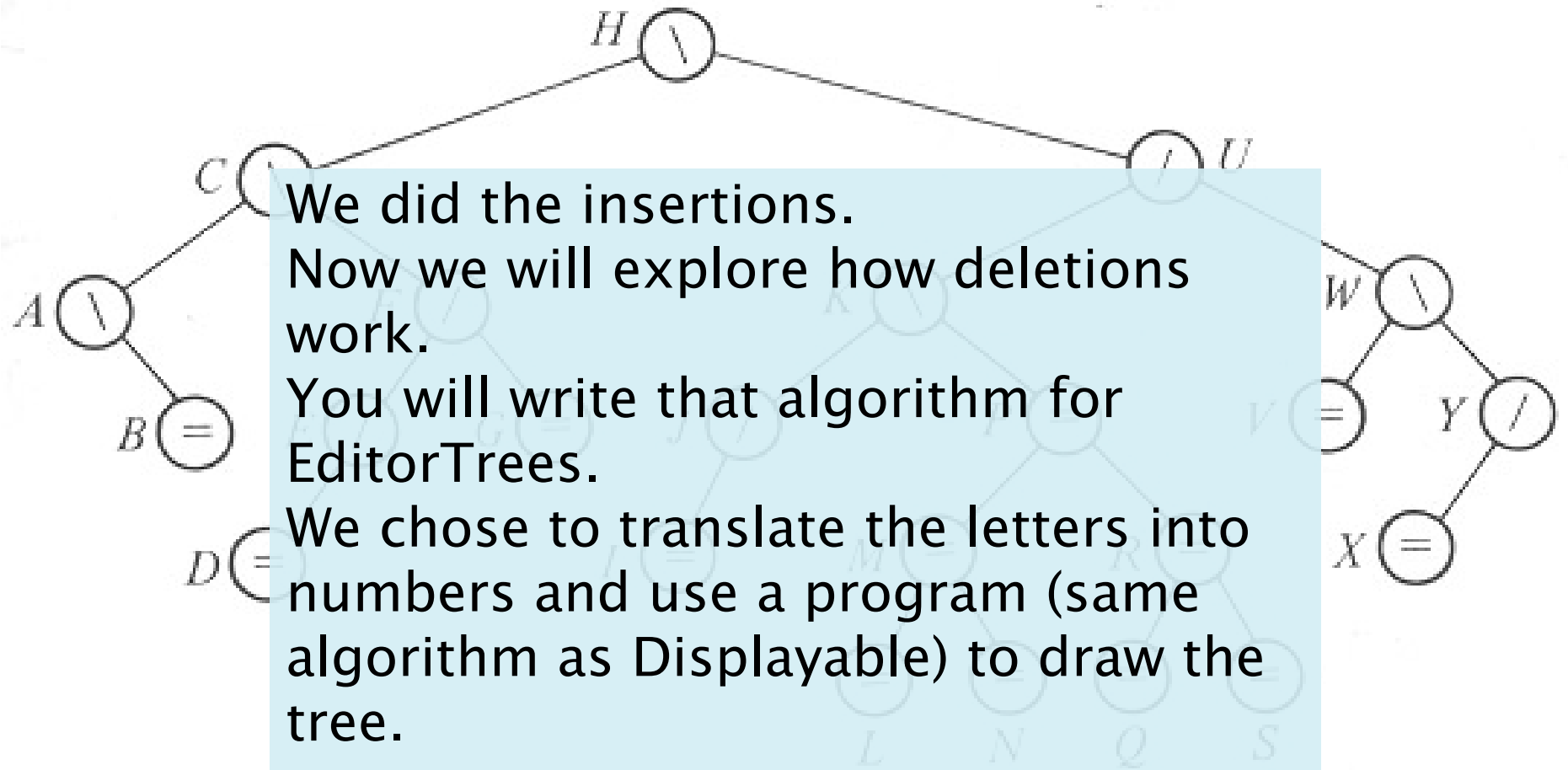
Overview of "rebalance after insertion"

- 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.
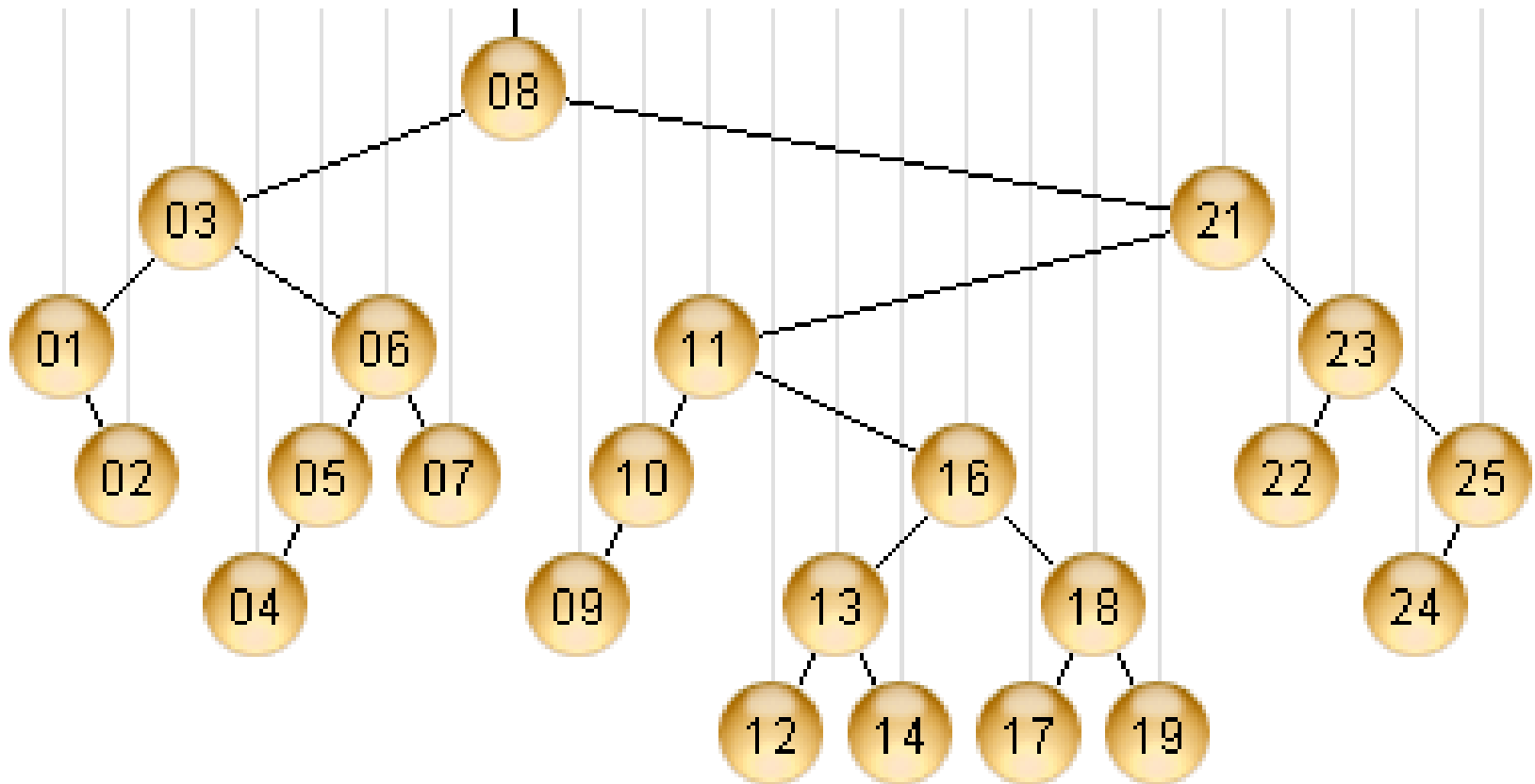   - do the appropriate rotation
    - break

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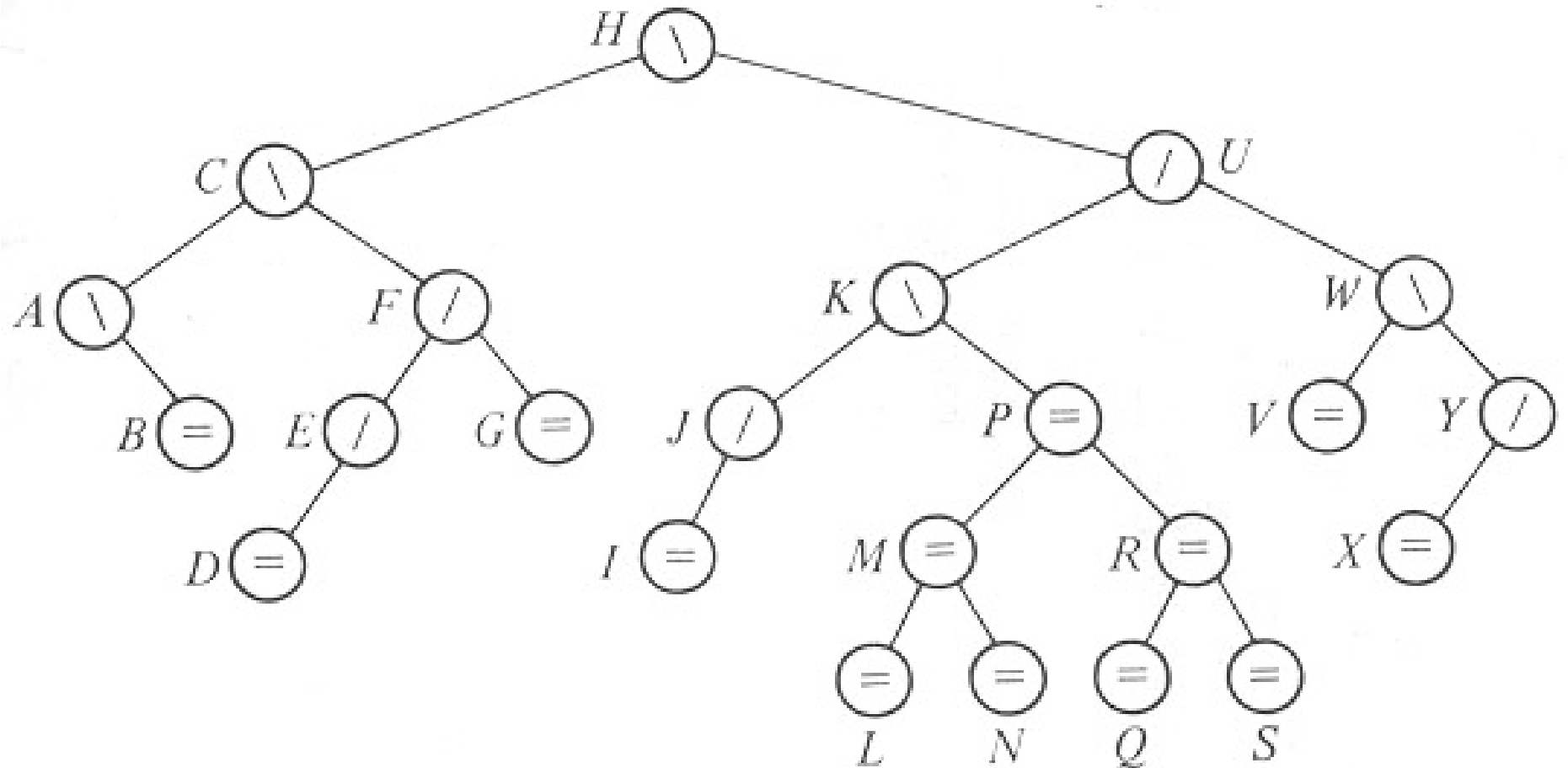
Insert **HA** into the tree, then **DA**, then **O**.
Delete **G** from the original tree, then **I, J**, **V**.

A sample AVL tree

We did the insertions.
Now we will explore how deletions work.
You will write that algorithm for EditorTrees.
We 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**.
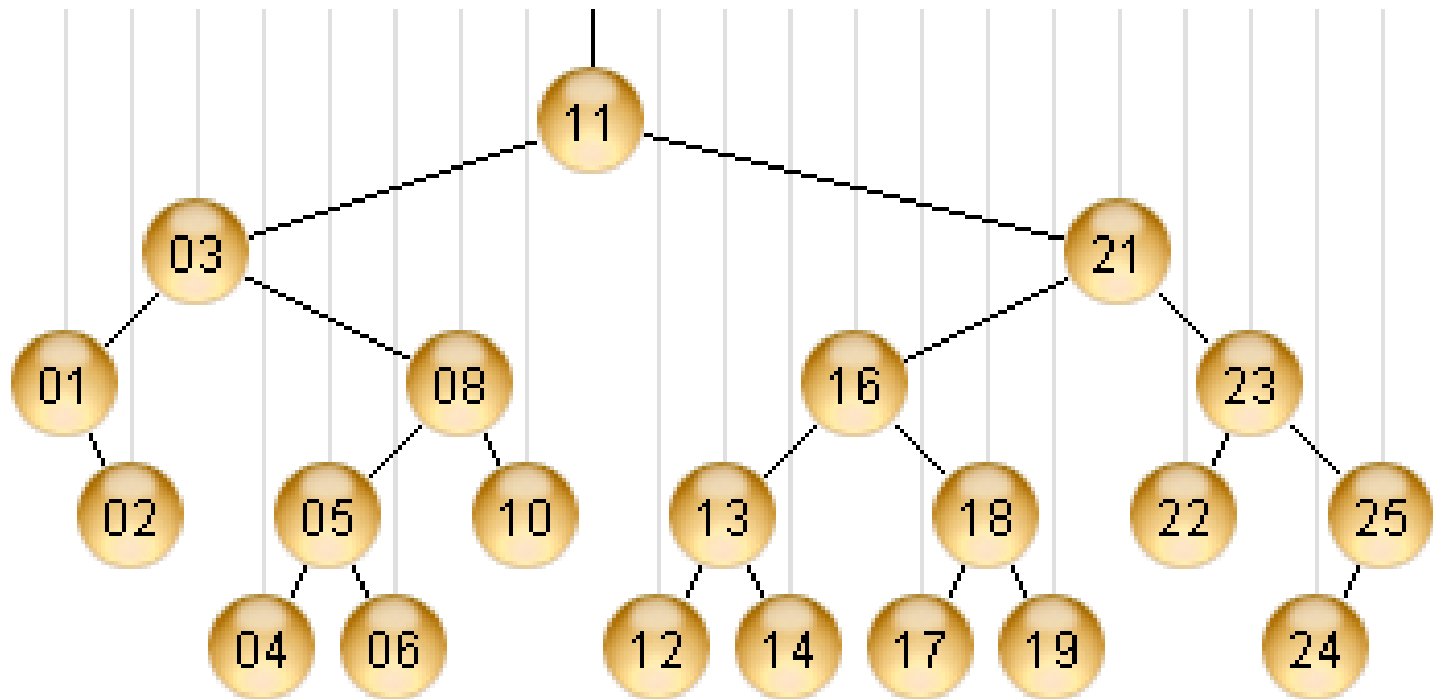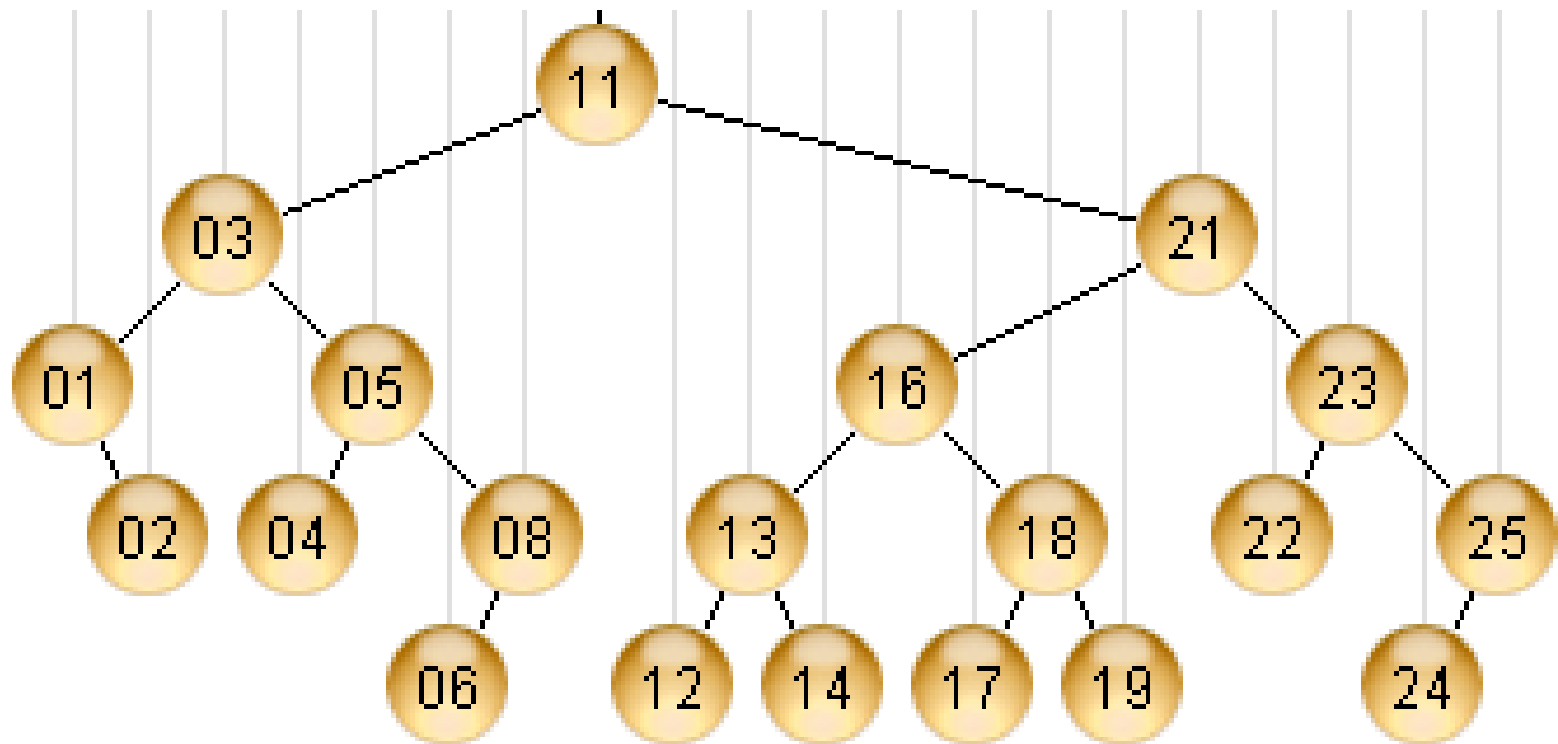Delete **G** from the original tree, then **I, J**, **V**.

Delete 7 and Rebalance

http://webdiis.unizar.es/asignaturas/EDA/AVLTree/avltree.html
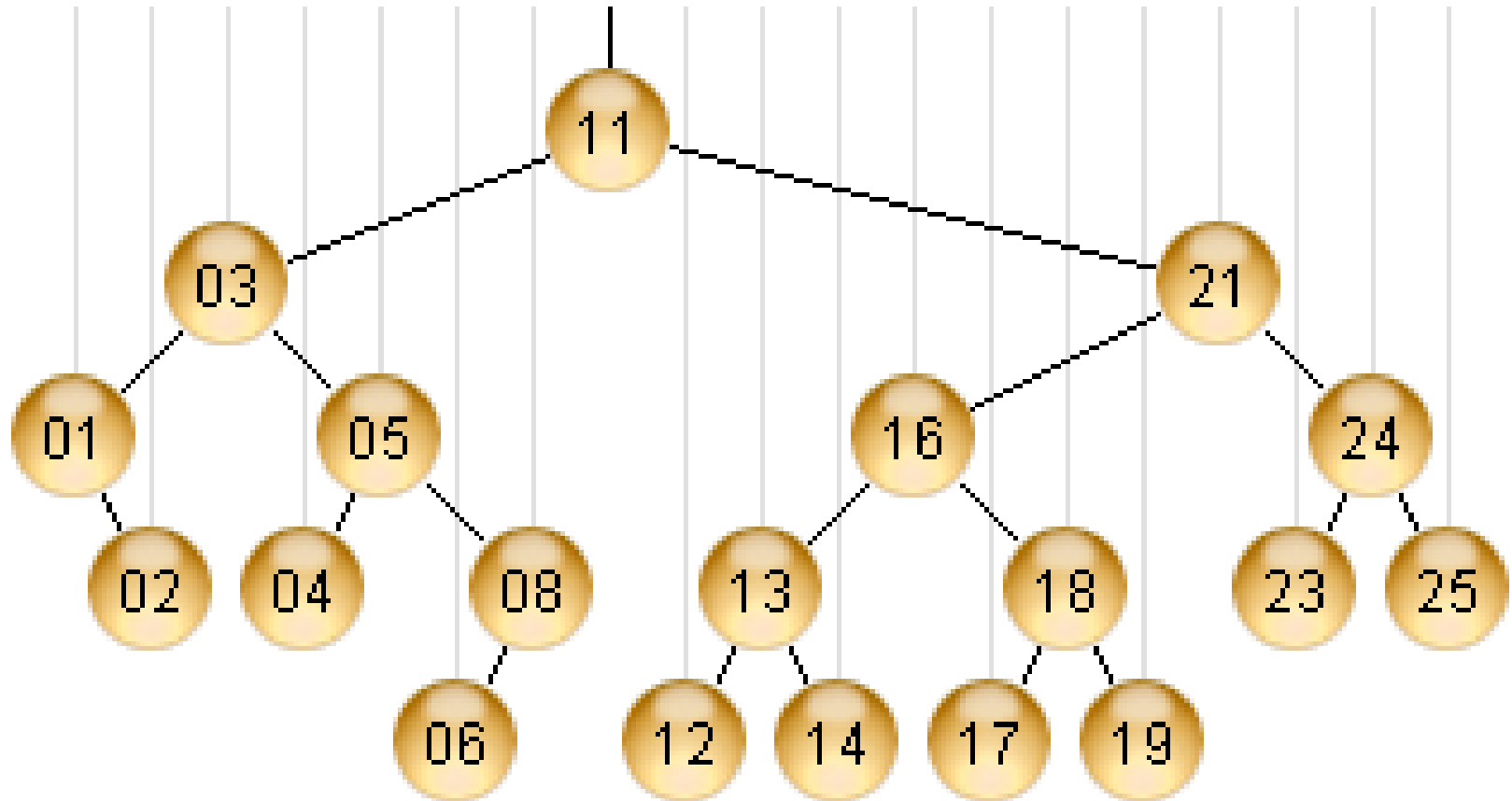
Delete 9 and Rebalance
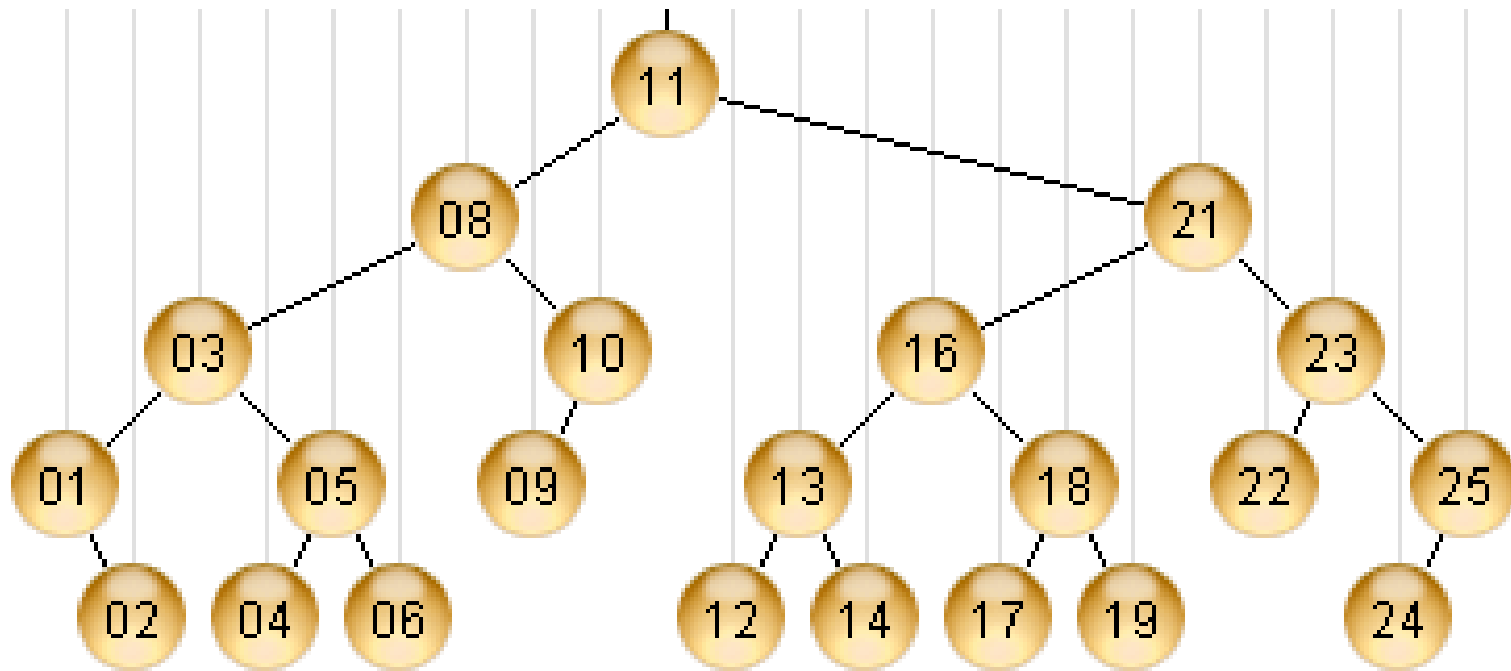
Delete 10 and Rebalance

Delete 22 and Rebalance

Final result

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 write the rotateDoubleRight code from yesterday's slide.