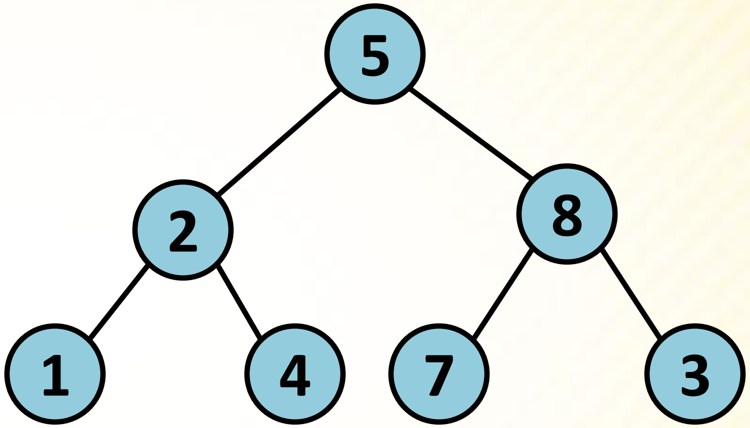
AVL TREES

COP 3502

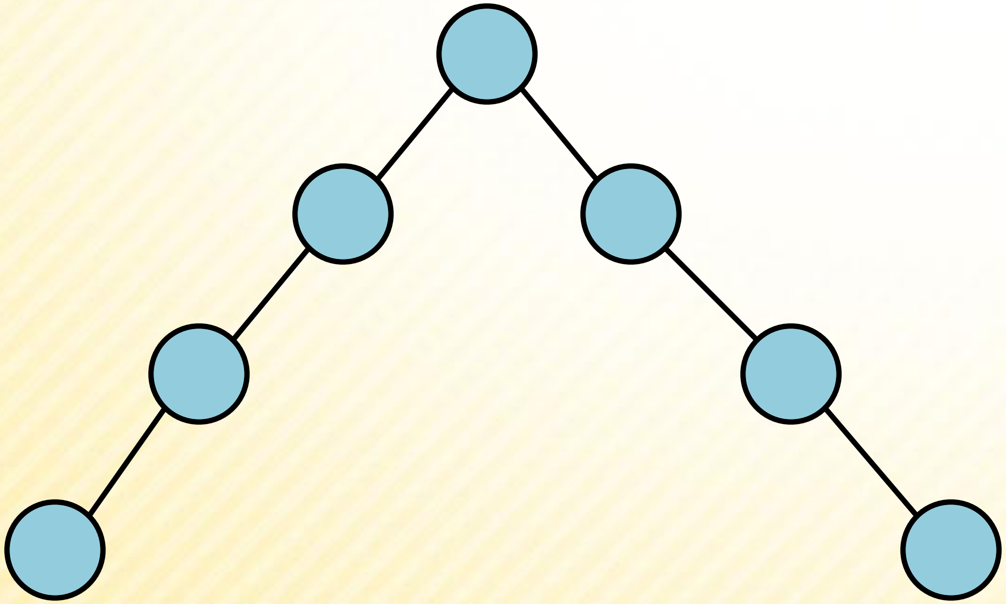
AVL Trees

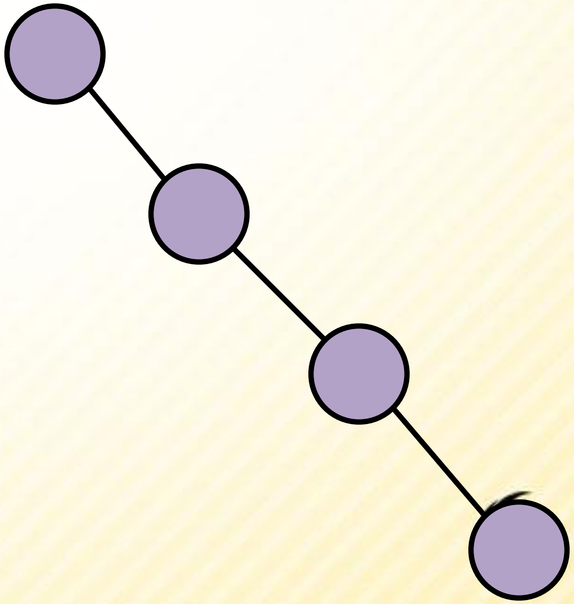
- We know that the search time for a node in a balanced binary search tree is O(log n)
  - We're dividing our search space in half each time we search the left or right branch.



AVL Trees

- But if trees get out of balance, or have deep search paths
  - Their search performance deteriorates
  - In the worst case instead of having an ***O(log n)*** search time
  - The search time is ***O(n)***



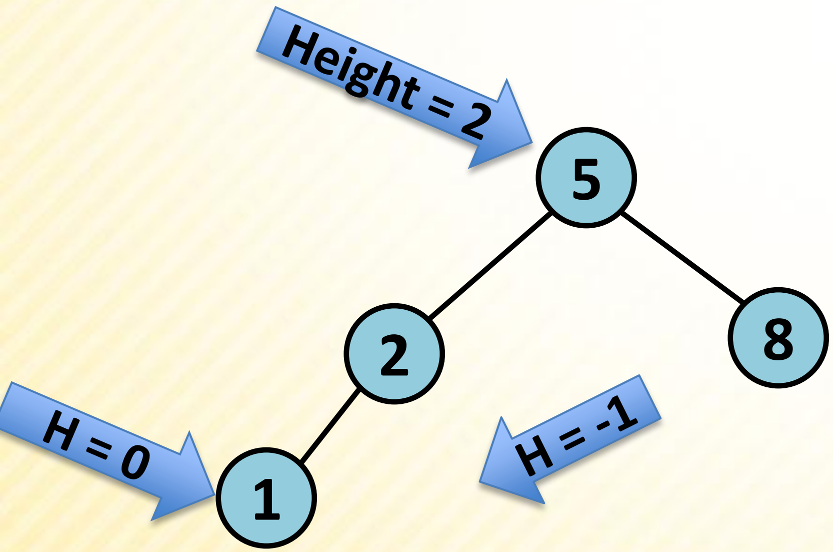


AVL Trees

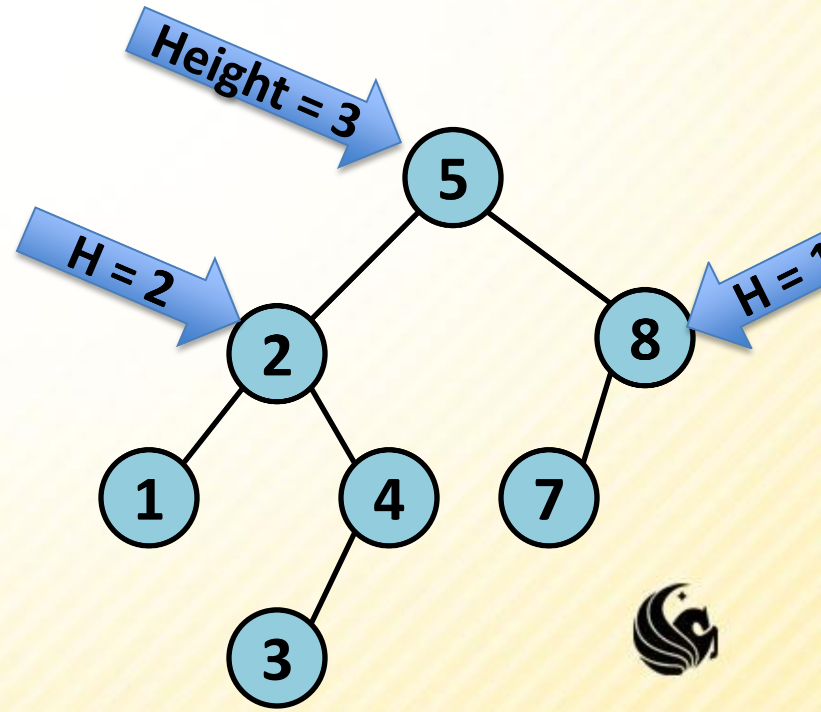
- So what we want is a tree that stays relatively balanced so that we can maintain the O(log n) search time,
  - BUT doesn’t require too much work in maintaining the balance so that we can still have O(log n) insertion time.
- 2 Russian mathematicians, Adelson-Velski and Landis, created this type of almost balanced trees –
  - known as AVL trees

AVL Trees

The **AVL tree property** is that for any node N, the height of N's left and right subtrees must be equal or differ by 1.



height of a binary tree:
the length of the longest path from the root to a leaf.
(the height of an empty tree is -1)
(the height of a leaf is 0)

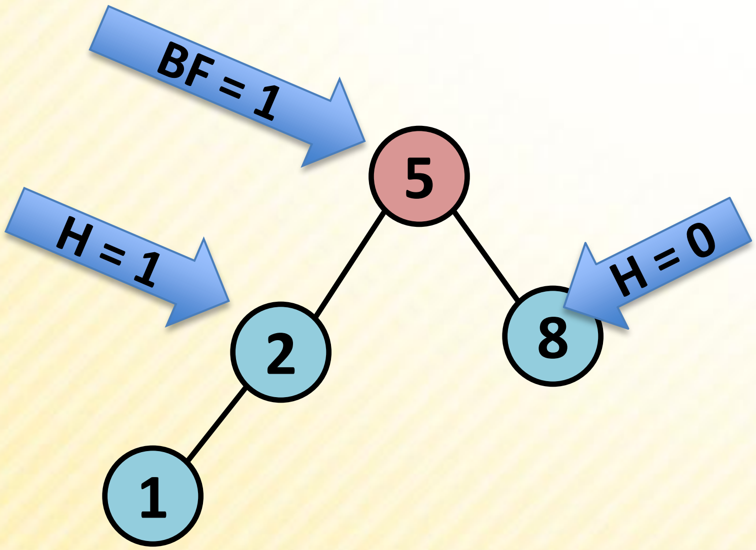


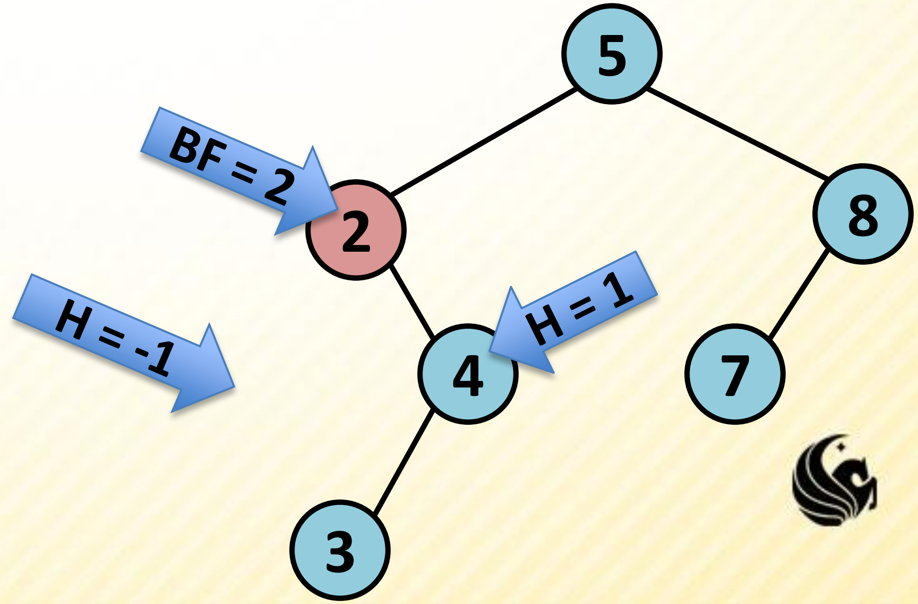
AVL Trees

The **AVL tree property** is that for any node N, the height of N's left and right subtrees must be equal or differ by 1.

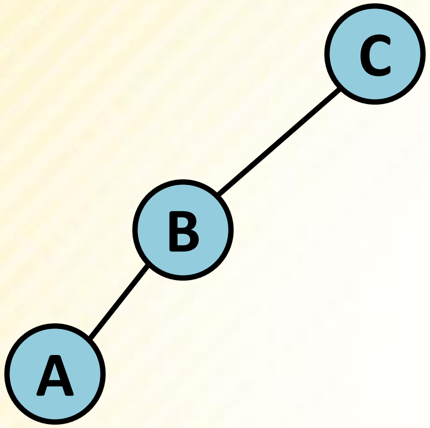
The **Balance Factor** is the difference in heights of the left and right subtrees at any node.

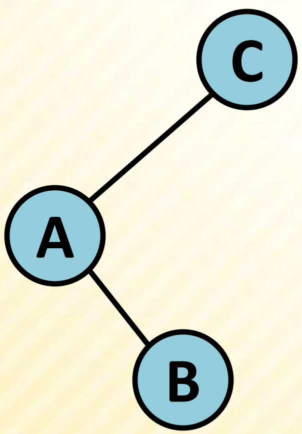
height of a binary tree:
the length of the longest path from the root to a leaf.
(the height of an empty tree is -1)
(the height of a leaf is 0)

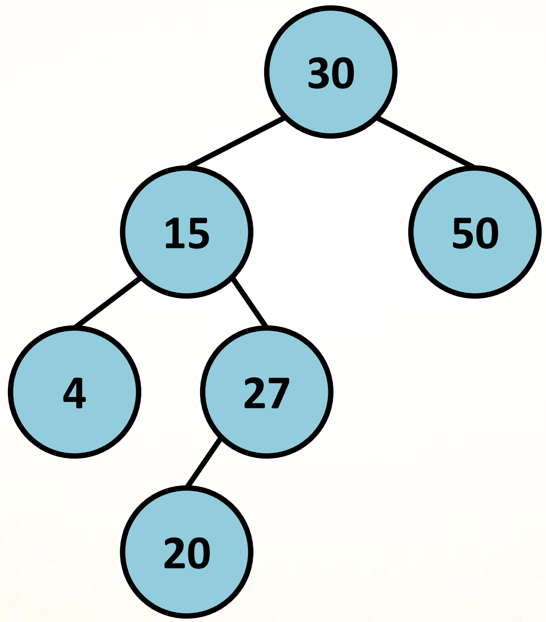


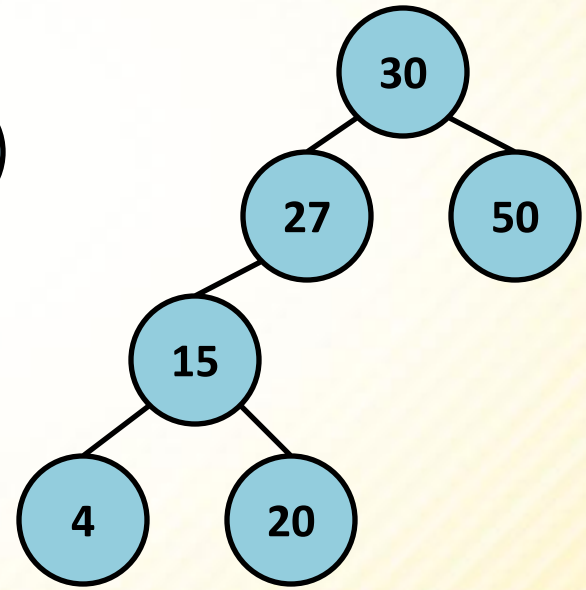


Non-AVL Trees



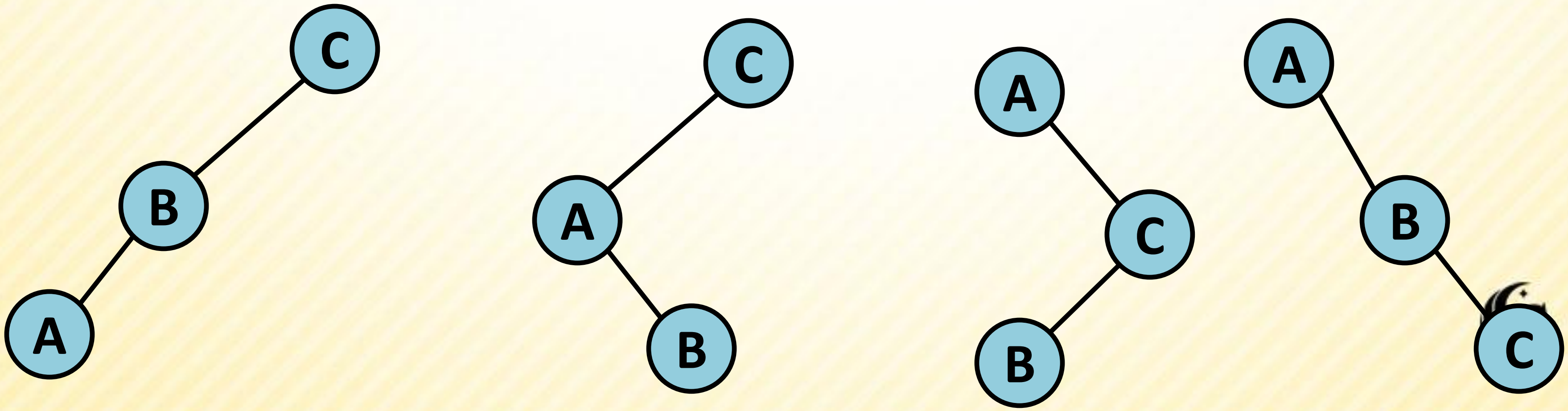






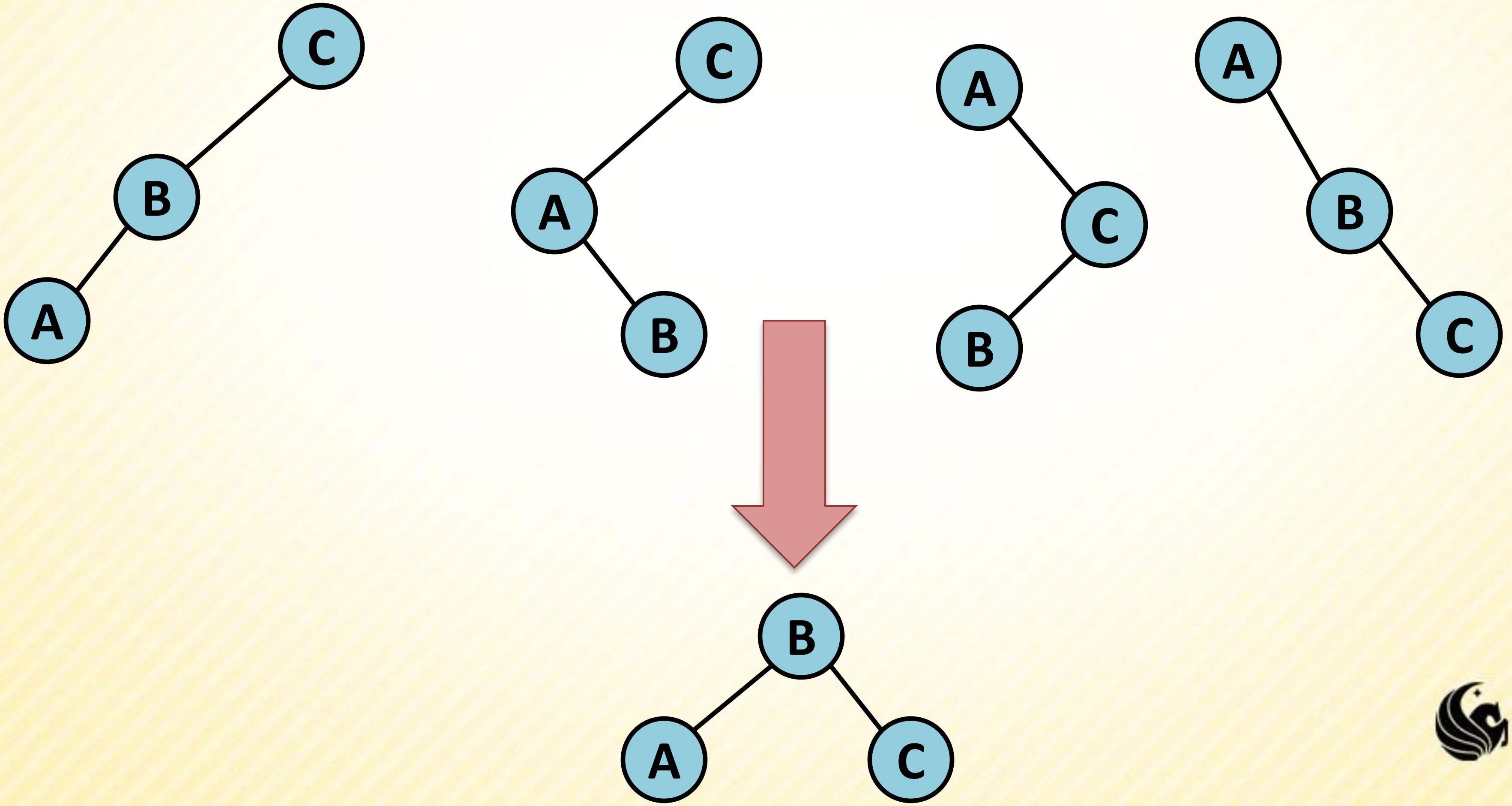
AVL Trees

- Now that we know what an AVL tree is,
  - now the question is how do we maintain this AVL tree property when new nodes are inserted or deleted?
  - When an imbalance is introduced to a tree, it is localized to 3 nodes and their 4 subtrees.
    - Denote the 3 nodes as **A, B, C** in their inorder listing.
    - Here are the 4 possibilites of the imbalances that could occur:



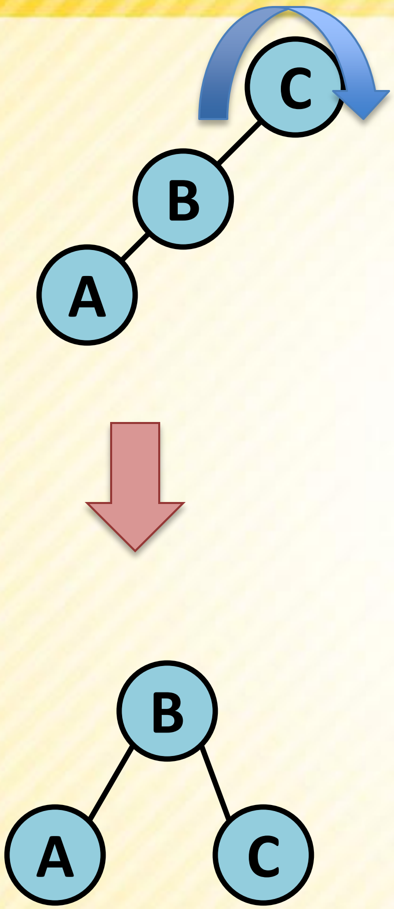
AVL Trees

- All 4 imbalance cases can be solved by converting to the following tree:



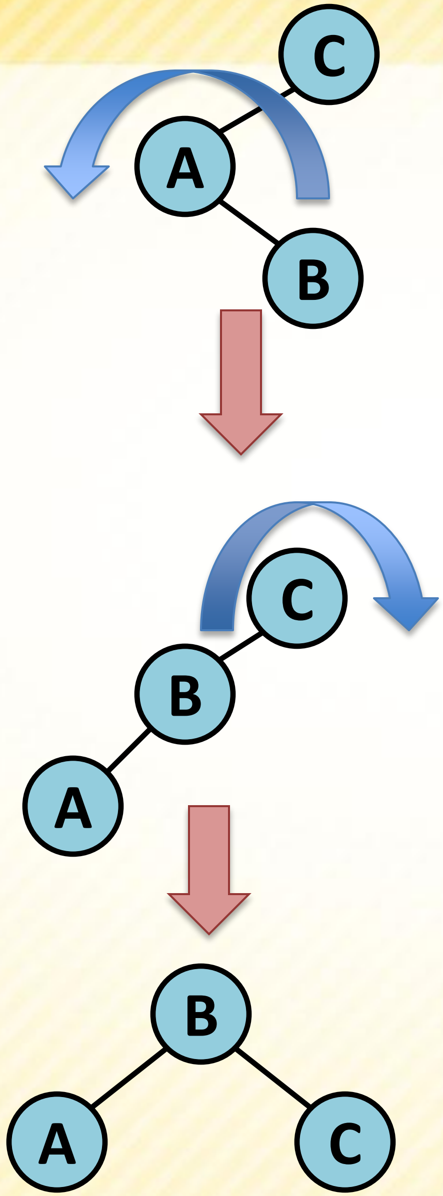
Case 1:

Single Rotation



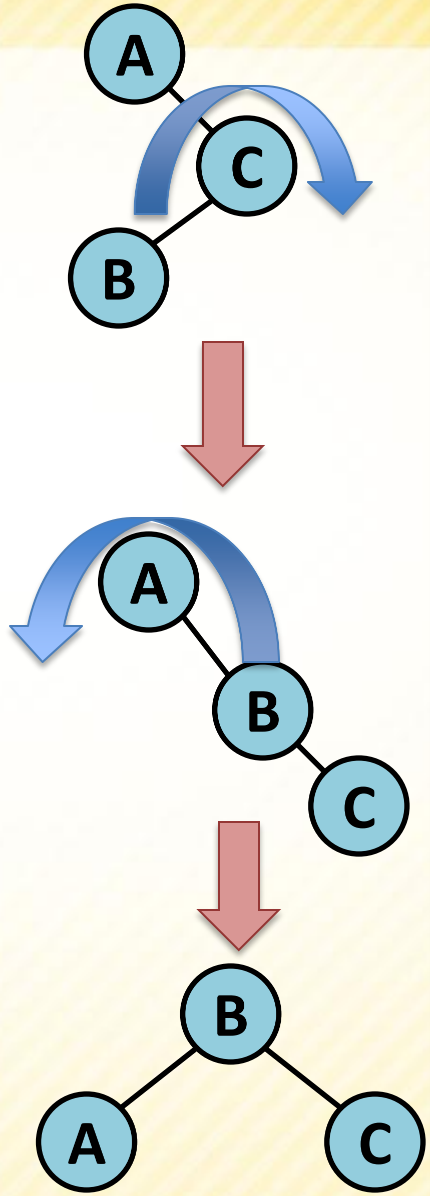
Case 2:

Double Rotation



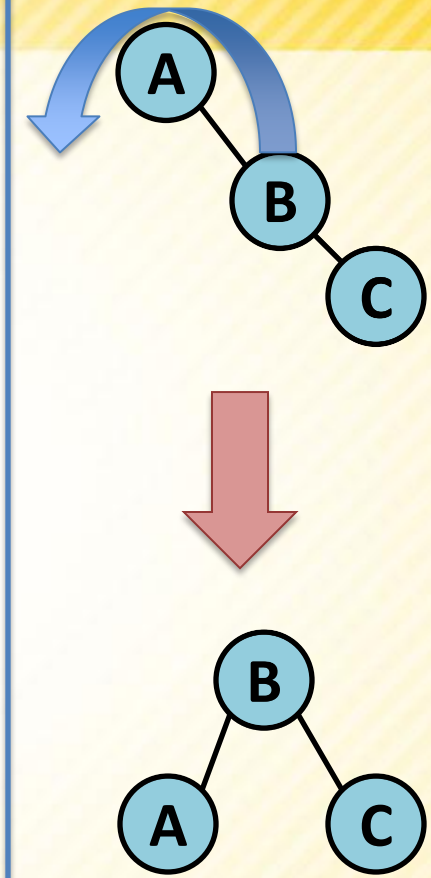
Case 3:

Double Rotation



Case 4:

Single Rotation

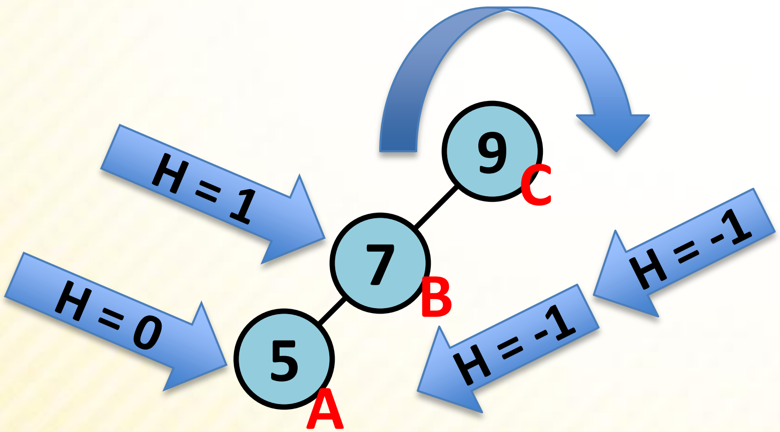


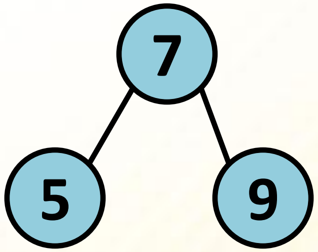
AVL Tree Insert

- So now the question is, how can we use these rotations to actually perform an insert on an AVL tree?
  - Here are basic steps:
    - Do a normal binary search tree insert
    - Restore the tree based on this new leaf node, steps for restoration:
      1) Calculate the heights of the left and right subtrees, use this to set the potentially new height of the node.
      2) If they are within one of each other, recursively restore the parent node.
      3) If not, then perform the appropriate rotations on that particular node, THEN recursively restore the heights of the parent node.
    - **Note:** No recursive call is made if the node in question is the root node and has no parents.
    - **Note:** one rebalancing will always do the trick, though we must make the recursive calls to move up the tree so that the heights stored at each node are properly recalculated.

AVL Tree Insert Examples

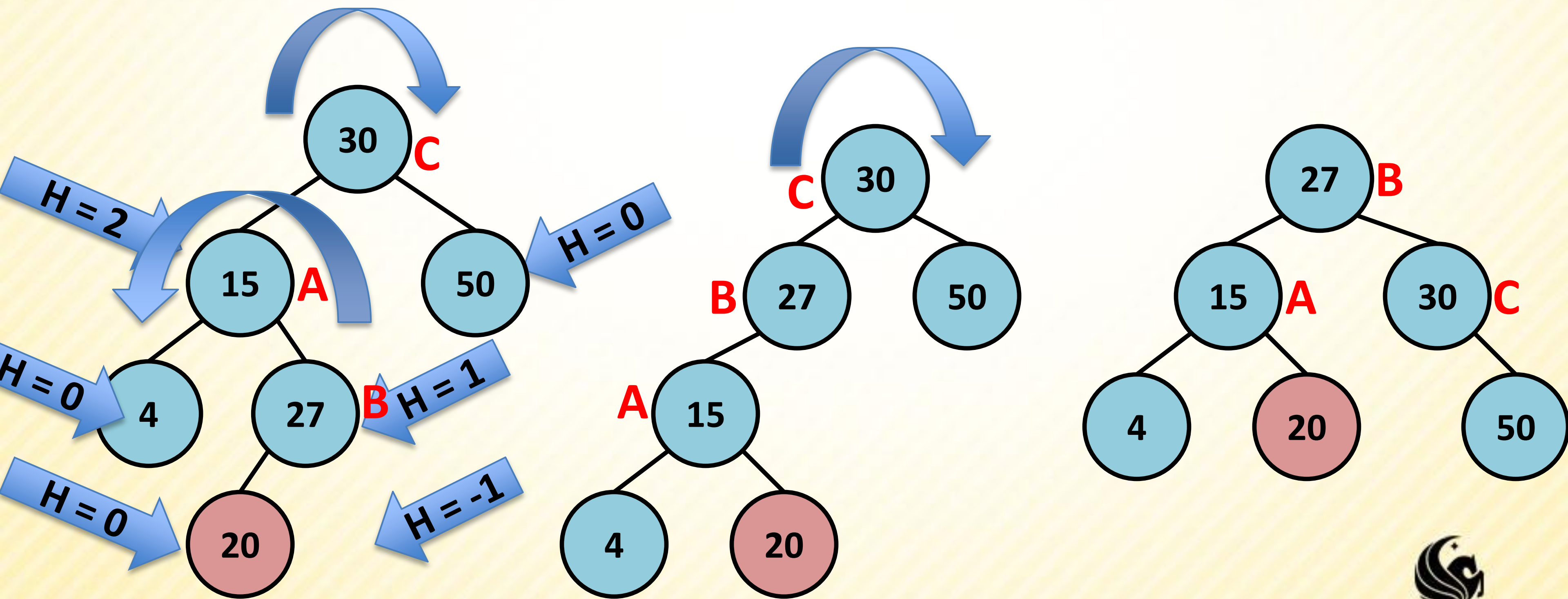
- The most simple insert into an AVL Tree that causes a rebalance is inserting a 3rd node into an AVL tree that creates a tree of height 2.
  - In this example, consider inserting the value 5:





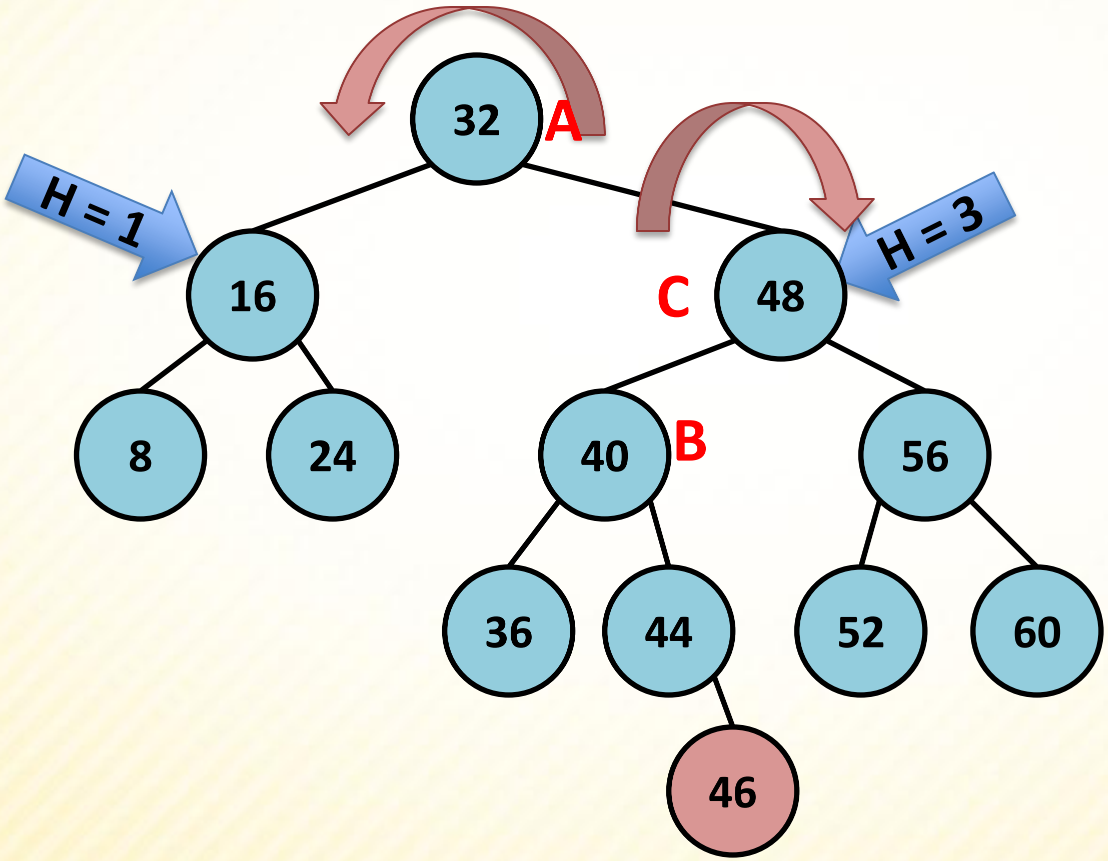
AVL Tree Insert Examples

- In this example, consider inserting the value 20:
  - **In this situation, the nodes 27 and 15 are balanced and we don't discover an imbalance until we trace up to 30. At this point, we label the nodes A, B and C based on our trace up the tree. The three values we passed were 27, 15 and 30, respectively. Thus, our labels are A = 15, B = 27, and C = 30.**



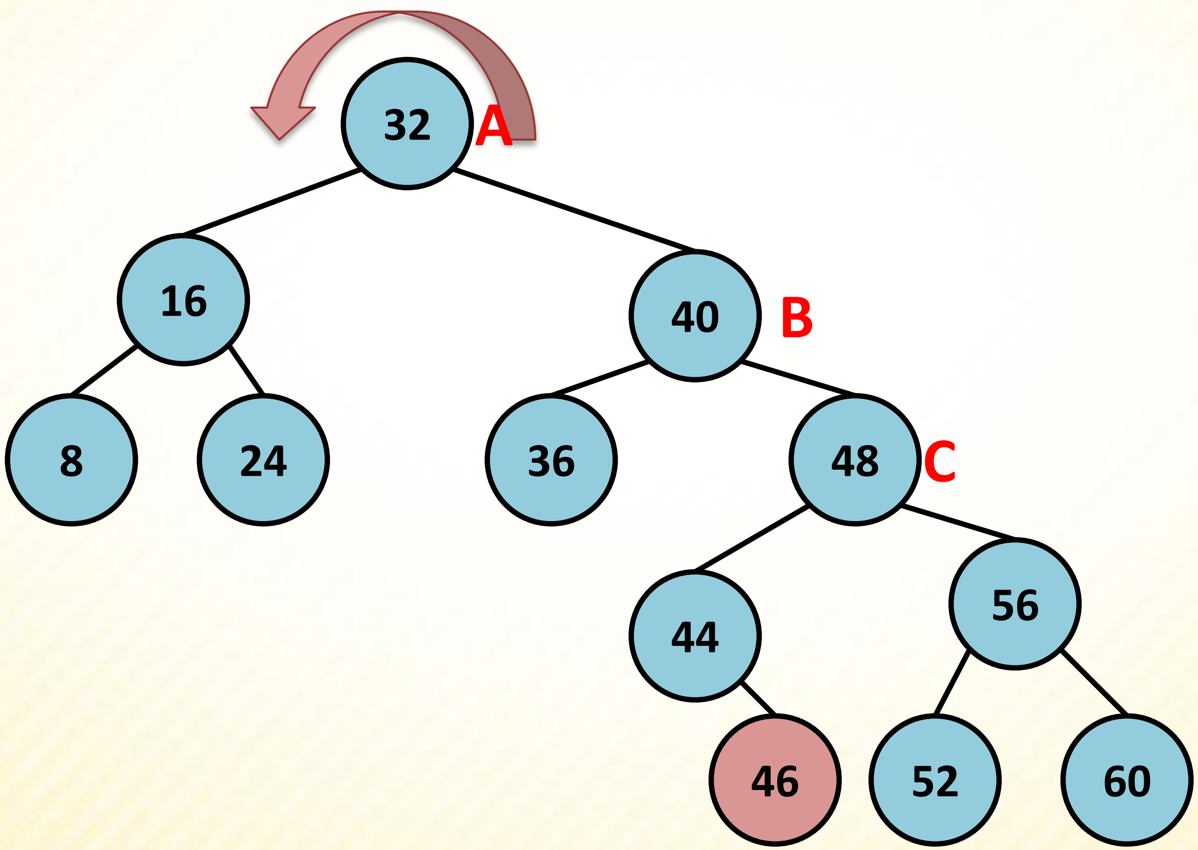
AVL Tree Insert Examples

- In this example, consider inserting the value 46:



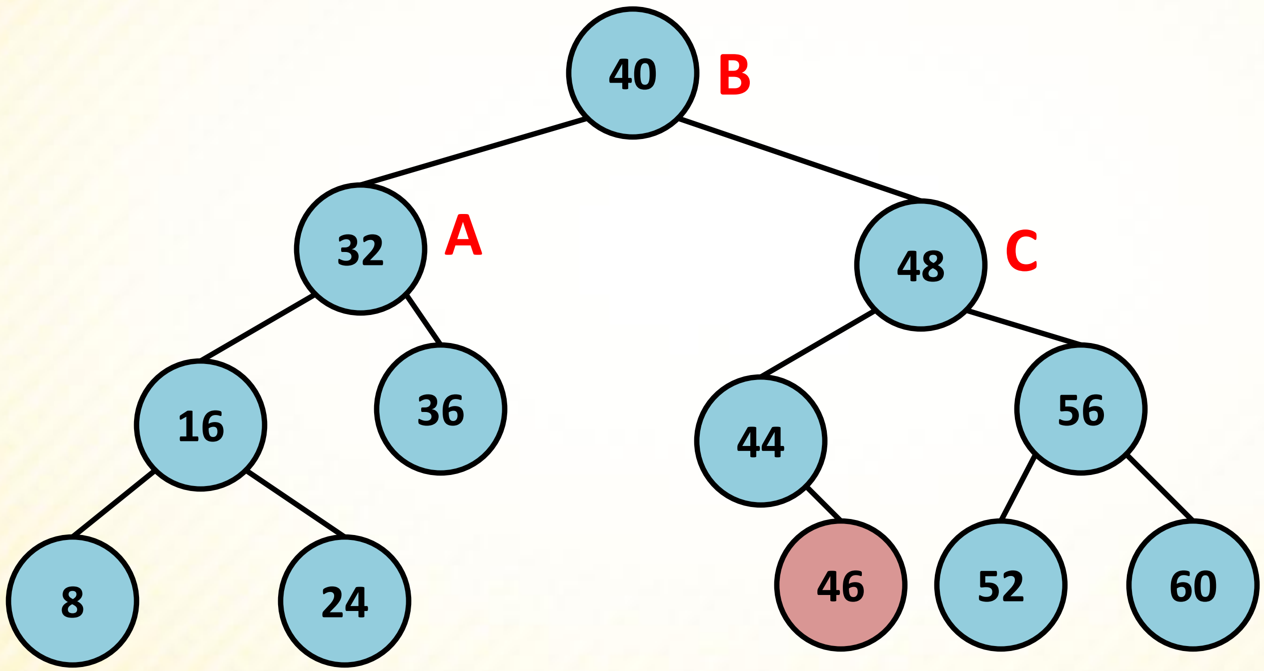
AVL Tree Insert Examples

- In this example, consider inserting the value 46:



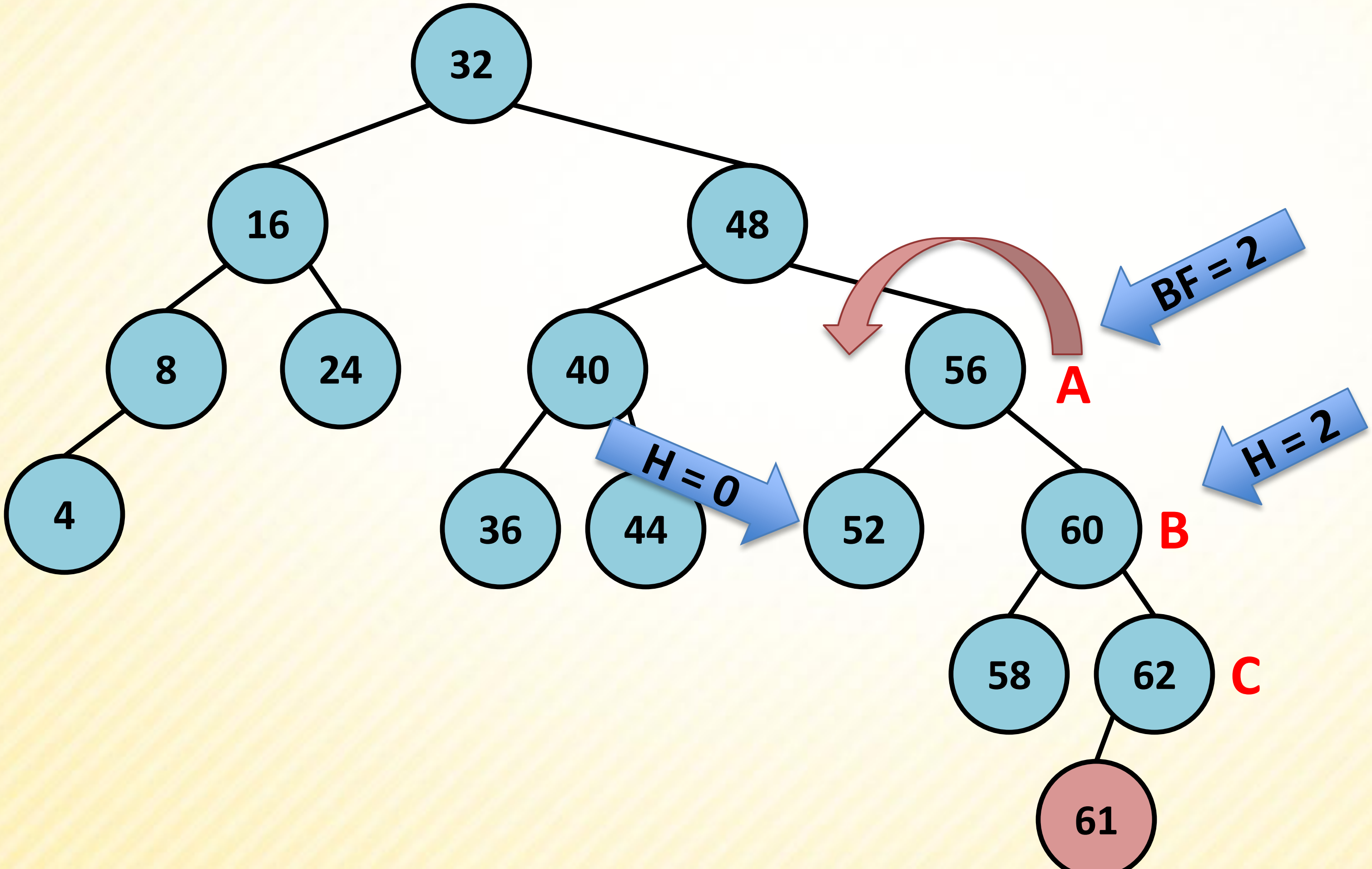
AVL Tree Insert Examples

- In this example, consider inserting the value 46:



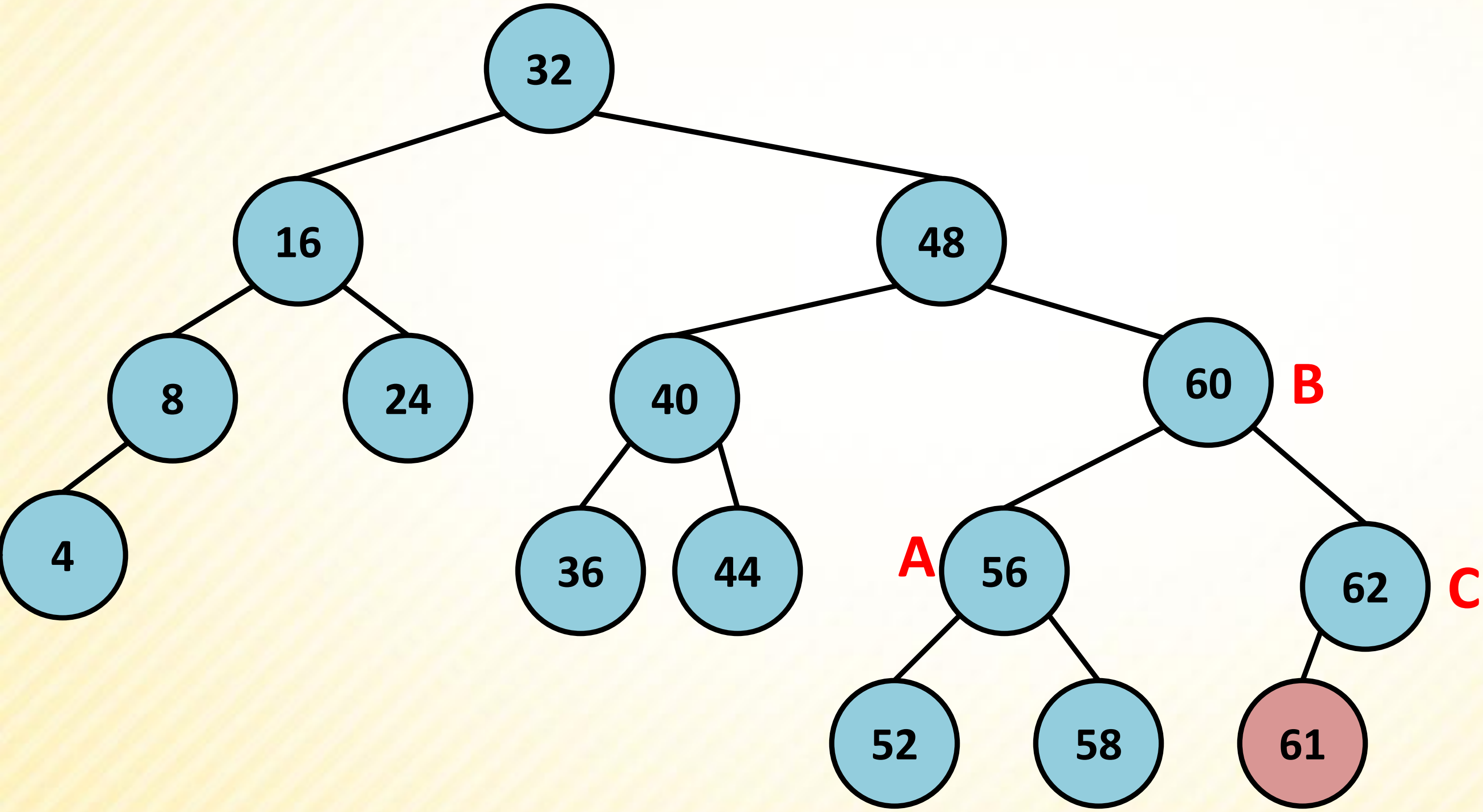
AVL Tree Insert Examples

- In this example, consider inserting the value 61:



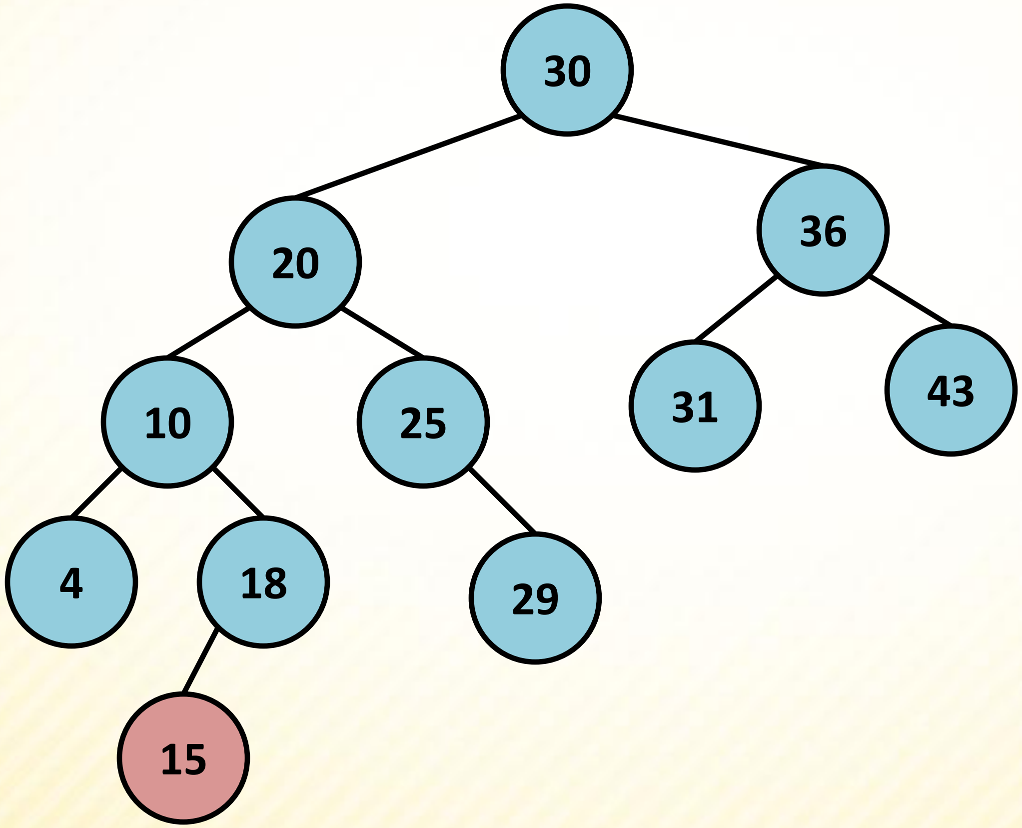
AVL Tree Insert Examples

- In this example, consider inserting the value 61:



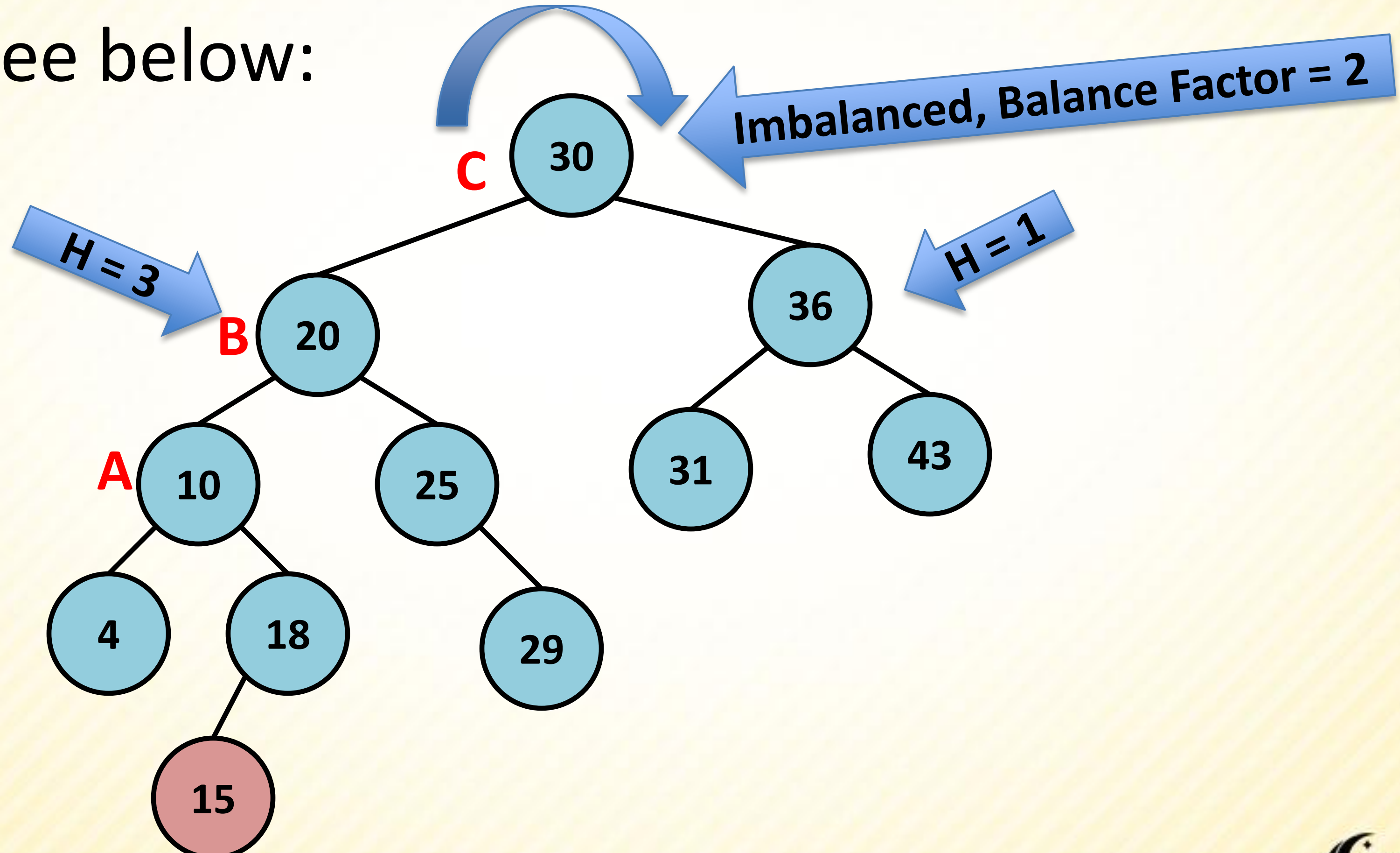
AVL Tree example

- Show the resulting tree after inserting 15 into the tree below:



AVL Tree example - ANSWER

- Show the resulting tree after inserting 15 into the tree below:



AVL Tree example - ANSWER

- Show the resulting tree after inserting 15 into the tree below: