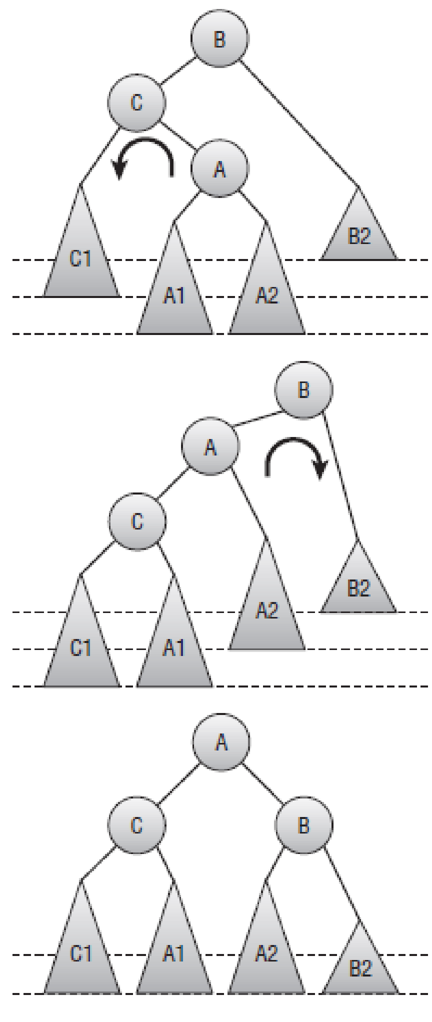
AVL Trees and 2-3 Trees



Agenda

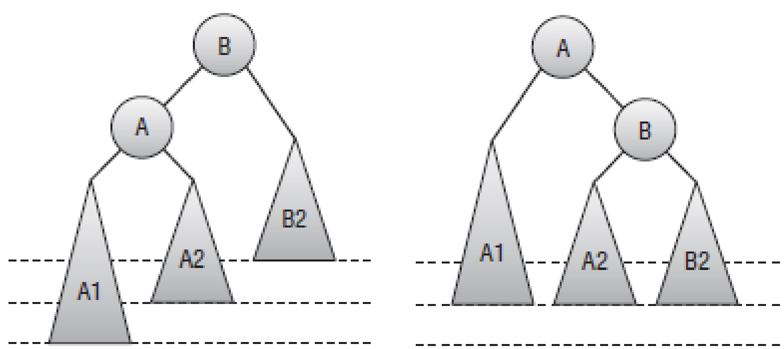
- AVL Trees
- 2-3 Trees
- Summary
- Exercises

AVL Trees

- A sorted binary tree
- The heights of two subtrees at any given node differ by at most 1

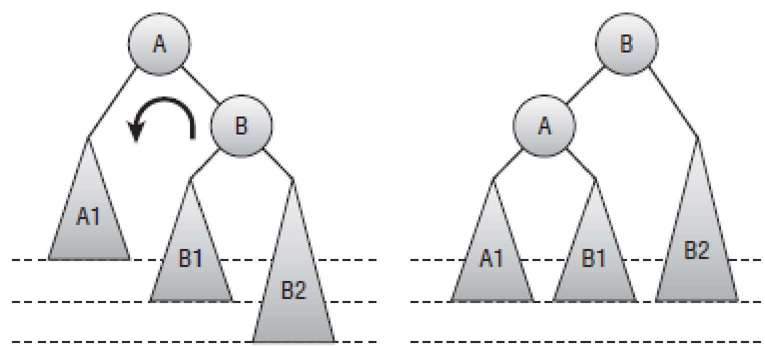
Adding Nodes, Left-Left Case

- Right rotation rebalances the tree



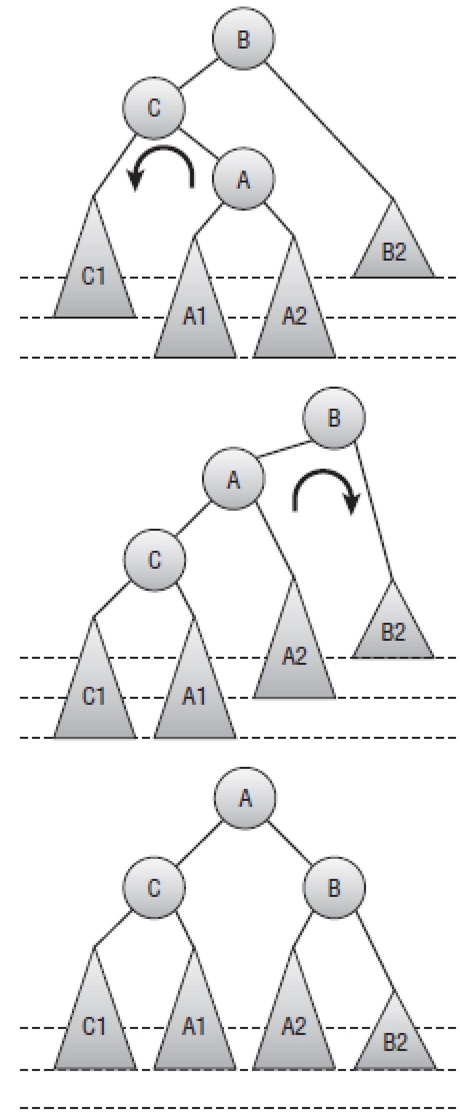
Adding Nodes, Right-Right Case

- Left rotation rebalances the tree



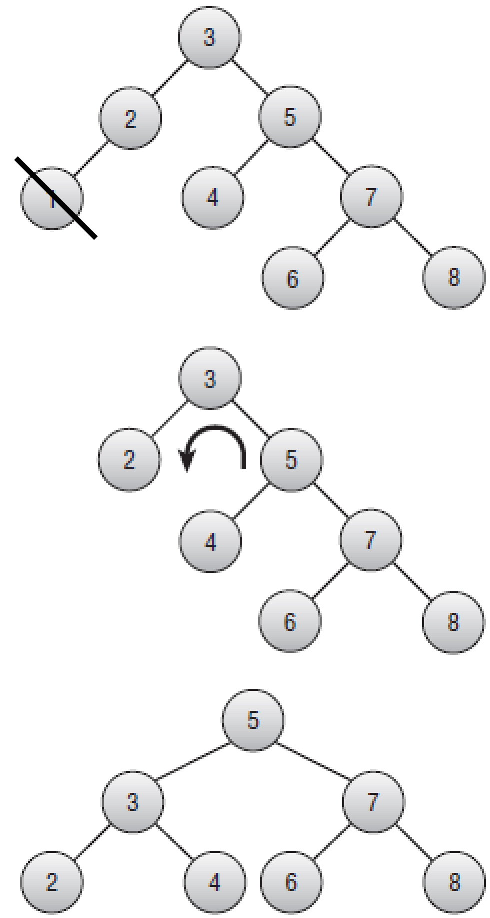
Left-Right Rotation

- A left rotation followed by a right rotation rebalances the tree if the new node is in the left child's right subtree



Deleting Nodes

- Use the same rotations

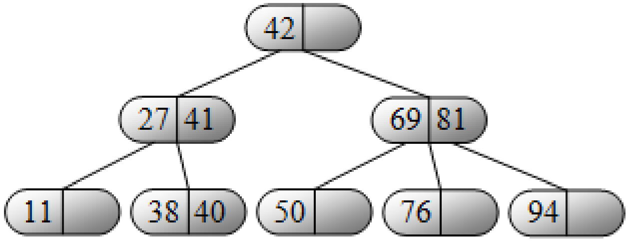


2-3 Trees

- Every internal node has either two or three children
- Nodes are called 2-nodes or 3-nodes
- Because every internal node has at least two children, a tree containing N nodes can have a height of at most $\log_2(N)$

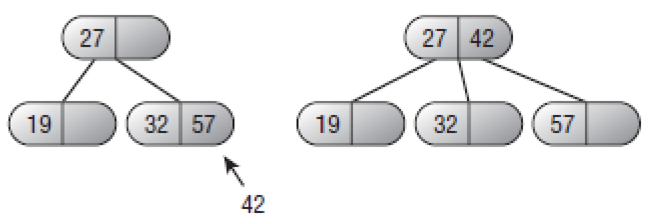
Finding Nodes

- Search down the appropriate branch



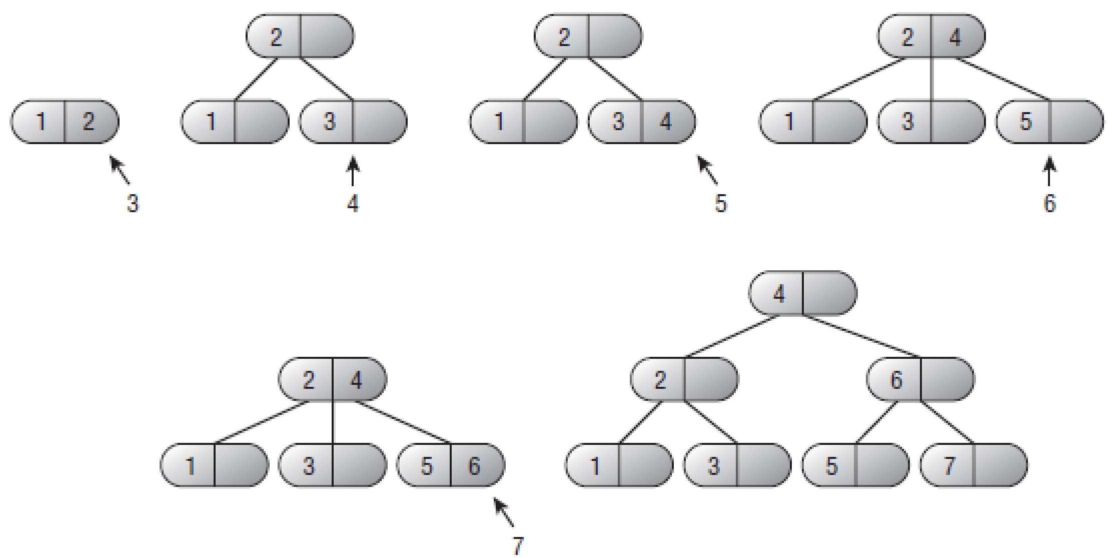
Adding to a Full Node

- Adding a value to a full node causes a node split



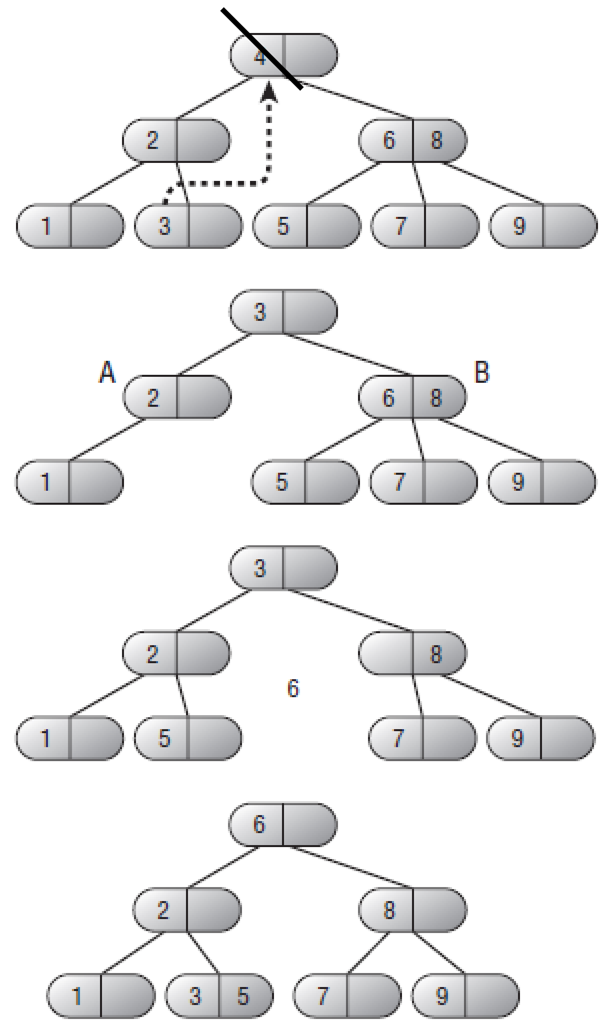
Adding Several Values

- Adding several values may cause many node splits
- 2-3 trees grow at the root



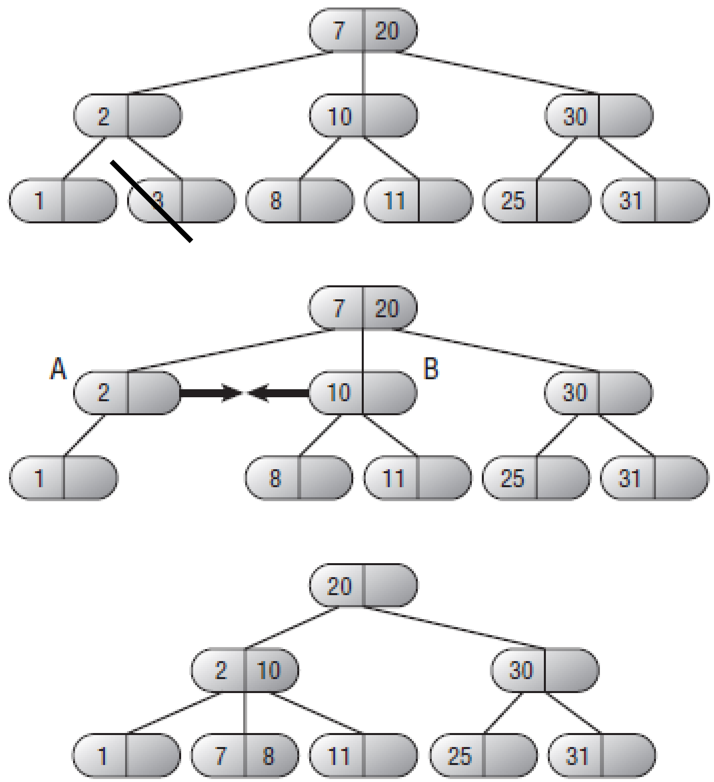
Deleting Values

- Replace with the rightmost descendant to the left
- Sometimes a node can borrow values from a sibling



Deleting Values (continued)

- Sometimes two nodes must merge



Summary

- [AVL Trees]
- [2-3 Trees]

Exercises

- Chapter 11 Exercises 1 – 5.
- Read *Essential Algorithms, 2e* Chapter 11 pages 359 – 366. (The rest of Chapter 11.)