COT 6405
ANALYSIS OF ALGORITHMS

Advanced Data Structure (B-Trees)

Computer & Electrical Engineering and Computer Science Department
Florida Atlantic University

Elementary Data Structures

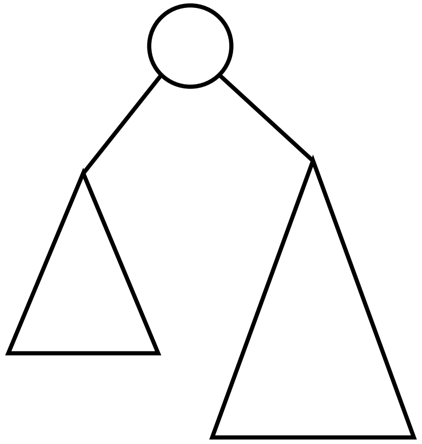
- Objective: data structure where the *dictionary operations* (insert, delete, search) take efficient RT
  - more specifically O(lg n)
- Elementary data structures
  - elementary data structures: stacks, queues, linked lists, hash tables, priority queues, binary search trees (BST) (ref. CLRS)
- Binary Search Trees (BST) :
  - all dictionary operations take O(h), where h – height of the tree
  - BST are not balanced with h = O(n)

Reference: *Introduction to Algorithms*, 3rd edition, by T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, The MIT Press, 2009.

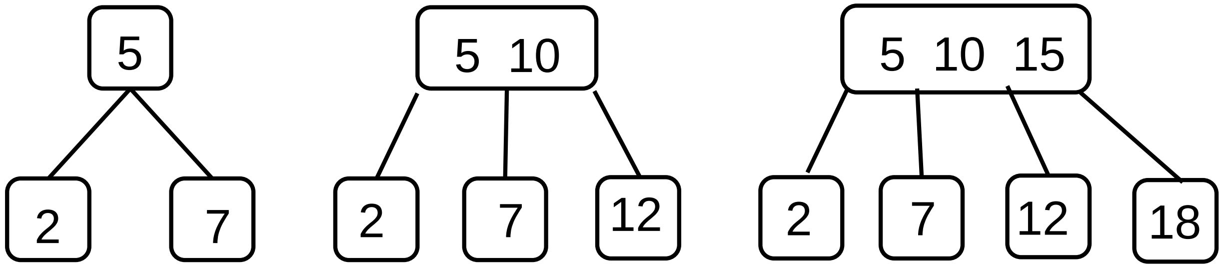
Balanced Search Trees, height h = $\Theta(\lg n)$

Two approaches:

- Transform an unbalanced BST to a balanced one
  - **AVL tree**: difference between the height of the left & right subtrees of a node never exceeds 1
  - **Red-black tree**: for any node, the height of a subtree is at most twice as large as the other subtree
  - If insertion/deletion destroys balance $\Rightarrow$ use ***rotations*** to restore the balance



- Representation change: allow more than one element in a node of a search tree
  - Perfectly balanced
  - 2-3-4 tree, B-tree



2-node 3-node 4-node

Next Steps

- Review BST (CLRS ch 12)
- Study B-tree (CLRS ch 18)