

Problem Set #2: Balanced Trees

Handed out on Feb. 26, due on Mar 4, at the beginning of class. Remember: write your own answers and use English or pseudocode when algorithms are requested. Late homeworks will not be accepted (turn in whatever you have).

Problem 0. In this problem you will show the state of an AVL tree following a series of operations. Draw the trees and label each node with its key and its balance factor.

- a. Start with an empty AVL tree. Show the state of the AVL tree after inserting each of the keys: 100, 150, 175, 125, 90, 99, 95 in that order. (You will draw 7 trees.)
- b. Starting from the last tree you found in problem 1a, show the *final* AVL tree after inserting the keys 87, 200, 110, 130, 135. (You will draw 1 tree.)
- c. Starting from the tree you found in problem 1b, show the result of deleting key 100. (You will draw 1 tree.)

Problem 1. AVL trees can be used to implement a List ADT, storing items i_1, i_2, \dots, i_n . The list items can be arbitrarily ordered and are not necessarily in sorted order. Explain how to store the items in an AVL tree and how to implement the operation $\text{access}(L, i)$ that returns the i th item in the List in time $O(\log n)$ for a list of n items. (Hint: store some information at each node about the sizes of the subtrees rooted there.)

Problem 2. Give a small binary tree T with a node labeled u such that $\text{splay}(T, u)$ operation increases the maximum depth of a binary search tree.

Problem 3. Show the result of $\text{splay}(T, i)$ on the tree at right:

Problem 4. Consider the B-tree of order 3 shown below:

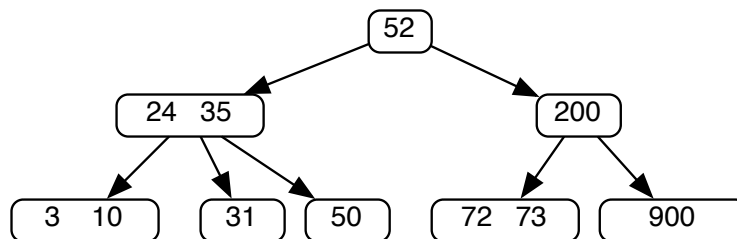
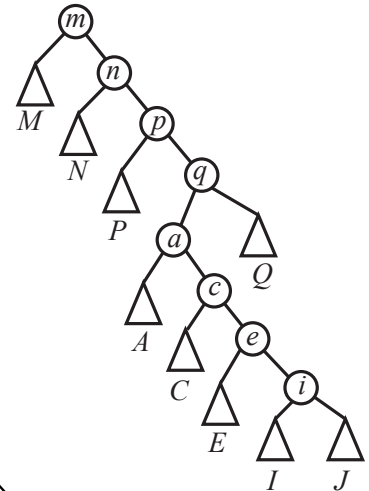


Figure 1: B-tree of order 3.

Show the state of the B-tree after each of the following sequences of operations (you will draw 3 trees). You should use key rotation when possible. For each part a,b, and c, start fresh with the tree in Figure 1.

- a. $\text{insert}(12)$
- b. $\text{insert}(55), \text{insert}(70), \text{insert}(150)$
- c. $\text{insert}(28), \text{insert}(33), \text{delete}(52), \text{delete}(73)$