

CSE 326 DATA STRUCTURES HOMEWORK 3

Due: **Friday, October 19, 2007** at the beginning of class.

Problem 1. Leftist Heaps

Leftist heaps support fast merges. This should mean that your organic processor should be able to do these operations efficiently!

- (a) Show the result of inserting keys 1 to 15 in order (i.e. 1 first, then 2 second, then 3 third, etc.) into an initially empty leftist heap. Use the leftist heap insert (i.e. merge) algorithm at each step. You don't need to show each step for this process, but be warned that if all you write down is the final answer and you get it wrong, it will be hard to award any partial credit. (Yes, this is a bit of busy work, but it should help get you started with part (b) of the problem.)
- (b) Prove or disprove: A perfectly balanced tree forms if keys 1 to $2^k - 1$ are inserted in order (again this means 1 first, then 2 etc) into an initially empty leftist heap. k is a positive integer. (Hints: induction; you have already worked through the base case above.)

Problem 2. Skew Heaps

Skew heaps have efficient amortized costs. If a step took a particularly long time, chances are that the next ones will be easy!

- (a) Weiss 6.26. You only need to show the final result, but note that if you do this it will be hard to award partial credit if the final result is incorrect.

Problem 3. Binomial Trees

A binomial tree of height 0, B_0 , is a one-node tree. A binomial tree of height k , B_k is formed by attaching a binomial tree, B_{k-1} to the root of another binomial tree another binomial tree B_{k-1} . (These are the same definitions as in Weiss.)

- (a) Weiss 6.32.
- (b) Prove that a binomial tree B_k has 2^k nodes.
- (c) Prove that a binomial tree of height k has $\binom{k}{d}$ nodes at depth d .

Problem 4. AVL Trees

AVL trees use rotations to make sure that left and right subtrees differ in heights by at most one. This ensures that the tree will not get inefficiently deep.

- (a) Show the result of inserting 2, 1, 4, 5, 9, 3, 6, 7 into an initially empty AVL tree.