CSE 326 DATA STRUCTURES
HOMEWORK 3

Due: **Monday, January 29, 2007** at the beginning of class in the lecture section you are registered in
Please put your quiz section (AA,AB,BA,BB) in addition to your name at the top of your homework

Problem 1. Leftist Heaps

Leftist heaps support fast merges. Hopefully this will make your own person running time in doing these problems short!

(a) Show the result of inserting keys 0 to 14 in order (i.e. 0 first, then 1 second, then 2 third, etc.) into an initially empty leftist heap. Note that you will be using the left heaps 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 you won't get any partial credit.

(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.

Problem 2. Skew Heaps Skew heaps have efficient amortized costs. Again this should make these problems less painful than they could have been!

(a) Weiss 6.26. You only need to show the final result, but note that if you do this you will have zero chance of getting partial credit.

(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 skew heap. k is a positive integer.

Problem 3. Binomial Trees

A binomial tree of hieght 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) Prove that a binomial tree B_k has 2^k nodes

(b) Prove that a binomial tree of height k has $\binom{k}{d}$ nodes at depth d.