CSE 326b DATA STRUCTURES
HOMEWORK 2

Due: **Friday, January 25** at the beginning of class. Please put your quiz section (BA, BB) in addition to your name at the top of your homework. As before, LaTeX pdf or other typeset documents are prefered, but not required.

Problem 1. Binary Min Heaps

This problem will give you some practice with the basic operations on binary min heaps.

(a) Starting with an empty binary min heap, show the result of inserting, in the following order, 10, 12, 1, 14, 6, 5, 8, 3, 9, 13, and 2, one at a time (using percolate up each time), into the heap. By *show* here we mean "draw the resulting binary tree with the values at each node."

(b) Now perform two `deleteMin` operations on the binary min heap you constructed in part (a). Show the binary min heaps that result from these successive deletions ("draw the resulting binary tree with values at each node").

(c) Instead of inserting the elements in part (a) into the heap one at a time, suppose that you use the linear time worst case algorithm described on page 211 of Weiss (Floyd's algorithm). Show the resulting binary min heap tree. (It would help if you showed the intermediate trees so if there are any bugs in your solution we will be better able to assign partial credit, but this is not required.)

Problem 2. d-Heap Arithmetic

Binary heaps implemented using an array have the nice property of finding children and parents of a node using only multiplication and division by 2 and incrementing by 1. This arithmetic is often very fast on most computers, especially the multiplication and division by 2, since this corresponds to simple bitshift operations. In d-heaps, the arithmetic is also fairly straightforward, but is no longer necessarily as fast. In this problem you'll figure out exactly what this math is.

(a) Weiss problem 6.14.

Hint: There are fairly concise solutions to this problem; if your solution is becoming particularly complicated, you might want to rethink your approach. In particular, it is worth thinking about where to put the root in the array, as some choices may simplify the calculations compared to others.

Problem 3. Min-Max Heaps

One problem with binary min heaps is that finding the maximum element in the heap cannot be done in logarithmic time. A solution to this is to use what is called a Min-Max heap. In this problem you'll explore the min-max heap and give algorithms for insertion and deletion of the min and max into the min-max heap.

(a) Weiss 6.18 parts (a), (b), and (c).

For parts (b) and (c), your answers should be given using a sufficiently precise pseudo-code to make your algorithms clear, but they need not be a syntactically correct and legal program in Java or other programming language.