CSE 326: Homework #3

Due: Wednesday, July 13

1. Recommended: You should know how to do these problems, but don't need to turn in a solution. Weiss, 4.2, 4.9, 4.19.

2. Weiss, 4.6. Use induction for your proof.

3. Weiss, 4.37

4. In class we saw how to implement the binary tree data structure using pointers. The data structure can also be implemented using arrays. Consider storing a complete binary tree of size N in an array of size N. The nodes are laid out in level order in the array: the root is at index 1, its left and right children are at index 2 and 3, the left child of 2 is at 4, and so on, with each level in the tree written out left to right.

   This extends to non-complete trees by placing nodes at the index where they would be if the tree was complete. Non-existent nodes are given a special marker value in the array, say -1. For example, the tree on the right hand side of Figure 4.15 in Weiss, page 109, would be represented in an array of size $N = 15$ as:

   6 2 8 1 4 -1 -1 -1 -1 3 7 -1 -1 -1 -1

   (a) If a node is at position i in the array, where are the left and right children of i? Where is the parent of i? Give a sentence or two justification for your answers.

   (b) Give pseudocode for iterative implementations of the Search ADT operations `find`, `insert`, and `delete` using an array for the underlying binary tree data structure. Your methods should be of the form `find( array A, integer N, integer key ) => integer`.

   (c) How does the pointer implementation compare to the array implementation in terms of cache behavior? Your answer should depend on the size of the tree.