CMPS 101
Final Review Problems

1. Let T be a binary tree, and let $N(T)$ and $H(T)$ denote its number of nodes and height, respectively. Show that $H(T) \geq \lfloor \lg(N(T)) \rfloor$. (Hint: this was proved in the solutions to hw6.)

2. Let T be an almost complete binary tree on n nodes, and let h be an integer in the range $0 \leq h \leq \lfloor \lg(n) \rfloor$. Show that the number of nodes in T at height h is exactly $\left\lfloor \frac{n}{2^h} \right\rfloor - \left\lfloor \frac{n}{2^{h+1}} \right\rfloor$.

3. Trace HeapSort on the following arrays
   a. (9, 3, 5, 4, 8, 2, 5, 10, 12, 2, 7, 4)
   b. (5, 3, 7, 1, 10, 12, 19, 24, 5, 7, 2, 6)
   c. (9, 8, 7, 6, 5, 4, 3, 2, 1)

4. Draw the Binary Search Tree resulting from inserting the keys: 5 8 3 4 6 1 9 2 7 (in that order) into an initially empty tree. Write pseudo-code for the following recursive algorithms, and write their output when run on this tree.
   a. InOrderTreeWalk()
   b. PreOrderTreeWalk()
   c. PostOrderTreeWalk()

5. Let T be a Binary Search Tree and let x be a node in T. Prove that if x has no right child, and if x has a successor y, then y is the lowest ancestor of x whose left child is also an ancestor of x.

6. The predecessor of a node x in a Binary Search Tree is defined to be the node which is printed immediately before x in an InOrderTreeWalk(). Let T be a Binary Search Tree and let x be a node in T. Suppose x has no left child, and x has a predecessor y. State a characterization of the predecessor y similar to the characterization of successor in problem 5. Write an algorithm called TreePredecessor() that returns the predecessor of a non-nil node x, if it exists, and returns nil otherwise.

7. State the Binary Search Tree Properties.