CS2223
HW#3

DUE: Monday, November 15

Consider the data structure *search tree* (described on pp. 157→159 of our text) to implement the abstract data type *set*. Note that listing the nodes of a search tree in *inorder* yields a sorted list. Hence, we can sort a list of n objects, $x_1 ... x_n$ by the following:

SEARCHTREESORT($x_1 ... x_n$)
    MAKEEMPTYSEARCHTREE(T)
    for $k \leftarrow 1$ **to** n **do** INSERT(x_k,T)
    Output T in *inorder*

(*a*) (2 points) What is the smallest possible height of a search tree of n nodes?
(*b*) (2 points) What is the largest height possible for a search tree of n nodes? Describe an input permutation which yields this worst-case height.
(*c*) (4 points) Using Θ–notation, what is the worst-case time to execute the step

for $k \leftarrow 1$ **to** n **do** INSERT(x_k,T)

of the preceding algorithm?
(*d*) (8 points) Test the average-case execution time of the preceding algorithm in the following way:

- Program and test the algorithm. Provide evidence that your algorithm works correctly.
- For several values of n, time your program for a random set of n integers.
- Try to fit a curve to your pairs (n, t_n), where t_n is the time to sort n integers. For example, you could try curves such as $t_n = c*n^2$ and $t_n = c*n*\lg n$.