CS 535 Homework 4
Due: October 7 (W), in class.

Undergrads, please answer questions 1,2,3. Question 4 is bonus. Graduate students, please answer all four problems.

1. Given a tree T with n nodes, we discussed a simple algorithm for computing the depth of a node v in T. Starting at v, we simply climb up the tree until we reach the root; the runtime is $O(depth(v))$, which is $O(n)$ since the depth of v can be as large as $n-1$. This time around, design an algorithm that computes the depth of *all* the nodes of T in $O(n)$ time. That is, in the worst case, computing the depth of all nodes in T has the same time complexity as computing the depth of a single node in T.

2. Suppose the following numbers are already stored in $A[1 \ldots 5] = [50, 40, 30, 20, 10]$. Illustrate the performance of the *inplace* heap-sort algorithm if $A[1 \ldots 5]$ is its input.

3. C-2.23. Here's an example: the union of intervals $[0.5, 3], [1, 2], [2, 5.6], [6.1, 6.5]$ is another set of intervals $[0.5, 5.6], [6.1, 6.5]$.

4. C-2.32.