Analysis of Algorithms: Assignment 2

Due date: January 21 (Thursday)

Problem 1 (3 points)
Prove the following properties of asymptotic bounds:

(a) If $f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n))$, then $f(n) = \Theta(h(n))$.

(b) $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

(c) $f(n) = o(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) \neq \Theta(g(n))$.

Problem 2 (2 points)
Give an example of functions $f(n)$ and $g(n)$ such that $f(n) \neq O(g(n))$ and $f(n) \neq \Omega(g(n))$.

Problem 3 (2 points)
Suppose that we have four algorithms, called A_0, A_1, A_2, and A_3, whose respective running times are n, n^2, $\lg n$, and 2^n. If we use a certain old computer, then the maximal sizes of problems solvable in an hour by these algorithms are s_0, s_1, s_2, and s_3.

Suppose that we have replaced the old computer with a new one, which is k times faster. Now the maximal size of problems solvable in an hour by A_0 is $k \cdot s_0$. What are the maximal problem sizes for the other three algorithms, if we run them on the new computer?

Problem 4 (3 points)
Determine asymptotic upper and lower bounds for each of the following recurrences. Make your bounds as tight as possible.

(a) $T(n) = 2T(n/2) + n^3$.

(b) $T(n) = T(n-1) + n$.

(c) $T(n) = T(\sqrt{n}) + 1$.