Algorithms: Assignment 2

Due date: September 14 (Thursday)

Problem 1 (3 points)
Give an example of functions $f(n)$ and $g(n)$ that satisfy all of the following conditions:

$$f(n) = O(g(n))$$
$$f(n) \neq \Theta(g(n))$$
$$f(n) \neq o(g(n))$$

Problem 2 (4 points)
Give a precise mathematical proof of the following asymptotic bounds:

(a) $\sqrt{n} = o(n)$

(b) for any constant $a > 0$,
$(n+1)^a = \Theta(n^a)$

Problem 3 (3 points)
Prove the following transitivity property of asymptotic bounds:

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

Problem 4 (bonus)
This problem is optional, and it does not affect your grade for the homework; however, if you solve it, then you will get 2 bonus points toward your **final grade** *for the course. You cannot submit this bonus problem after the deadline.*

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?