CS 173: Discrete Structures, Fall 2009
Solutions to review problems for second quiz

1. **Big-O notation**

   For each of the following pairs of functions state whether $f(n) = O(g(n))$ or $f(n) = \Omega(g(n))$ or $f(n) = \Theta(g(n))$

   (a) $f(n) = \lceil n \rceil^2$ and $g(n) = \lfloor n \rfloor^2$.
   Solution: $f(n) = \Theta(g(n))$. The two functions are equal for integer inputs. For non-integer inputs, floor and ceiling can differ by at most 1. So the difference between the squares will be no more than $(n^2 + 2n + 1) - n^2 = 2n + 1$. As n gets large, this grows more slowly than n^2.

   (b) $f(n) = (log_{10}(n))^2$ and $g(n) = n$. The log function grows more slowly than n, and the square of the log function grows even more slowly.
   Solution: f(n)=O(g(n)).

   (c) $f(n) = n^{2^n}$ and $g(n) = n^{n^2}$, because 2^n grows much faster than n^2.
   Solution: $f(n) = \Omega(g(n))$.

   (d) $f(n) = n!$ and $g(n) = n^n$.
   Solution: $f(n) = O(g(n))$. If you expand both functions into products, their first terms are the same, but then the later terms are much smaller for $n!$. This difference gets bigger as n gets larger.

   (e) $f(n) = 2^n + n$ and $g(n) = 3^n$
   Solution: $f(n) = O(g(n))$. You can ignore the n term in f. As n gets large, the difference between 2^n and 3^n gets larger, because you keep multiplying in more factors of $\frac{3}{2}$.

   Determine whether each statement below is true or false.

   (f) If $f(n) = \Theta(g(n))$ and $h(n) = \Theta(g(n))$ then $f(n)h(n) = \Theta(g(n))$.
   Solution: False. $f(n)h(n) = 0(g(n)g(n)) \rightarrow \forall x > k, f(x)h(x) \leq cg(x)g(x)$. $cg(x)$ cannot be replaced by a constant.

   (g) If $f(n) = \Omega(g(n))$ and $h(n) = \Omega(g(n))$ then $f(n) + h(n) = \Omega(g(n))$.
   Solution: True. We have, $f(x) \geq c_1 g(x)$ for all $x > k_1$ and $h(x) \geq c_2 g(x)$ for all $x > k_2$. Hence we have that $f(x) + h(x) \geq c_1 g(x) + c_2 g(x) = (c_1 + c_2)g(x)$ for all $x > max(k_1, k_2)$.

   (h) If $f(n) = O(g(n))$ then $g(n) = \Omega(f(n))$.
   Solution: True. We saw this in lecture. (It's one way to define Ω.)

   (i) If $f(n) = \Theta(g(n))$ then $g(n) = \Theta(f(n))$
   Solution: True. Θ is a type of equality. If it's true in one direction, it's true in the other.

(j) If $f(n) = log_a(n)$ for $a > 2$ then $f(n) \neq \Theta(log_2(n))$.

Solution: False. Changing the base of $\log_a n$ multiplies all the outputs by a number that doesn't depend on n (it only depends on the old and new bases).

When the right answer is given as $f(n) = \Theta(g(n))$ above, you could also (optionally) say that $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.