ALGORITHMS, FALL 2018, HOMEWORK 1

Due Thursday, September 13 at 11:59pm.
Worth 1% of the final grade.
Submit each problem on a separate page. Subproblems can be on the same page.

1. For each of the following subproblems, there are two functions being compared. Call the first $f(n)$ and the second $g(n)$. Prove whether $f(n)$ is $O(g(n))$ or not, and also whether it is $\Omega(g(n))$ or not. Note that the requirement is to *prove*, not just *state*.
   (a) 3^{n+1} vs 3^n
   (b) 2^{2n} vs 2^n
   (c) 4^n vs 2^{2n}
   (d) n^2 vs $n^{2.01}$
   (e) $n^{0.9}$ vs 0.9^n.
2. Same instructions as problem 1.
   (a) $\log^c n$ vs $\log n$, where c is a constant greater than 1.
   (b) $\log n^c$ vs $\log n$, where $c = \Theta(1)$.
   (c) $\log(c \cdot n)$ vs $\log n$, where $c = \Theta(1)$. You can assume both have the same base.
   (d) $\log_a n$ vs $\log_b n$, where a and b are constants greater than 1.
   Show that you understand why this restriction on a and b was given.
3. Let $f(x) = O(x)$ and $g(x) = O(x)$. Let c be a positive constant.
   Prove or disprove that $f(x) + c \cdot g(y) = O(x + y)$.
4. Let $f(n) = \sum_{x=1}^{n} (\log^3 n \cdot x^{29})$. Find a simple $g(n)$ such that $f(n) = \Theta(g(n))$. (Prove both big-O and Ω). Don't use induction / substitution, or calculus, or any fancy formulas.
   Just exaggerate and simplify.