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

Let us assume that 3^{n+1} is $O(3^n)$ for all $n \geq 1$.

Therefore some constant c must exist $c < \infty$ such that $3^{n+1} \leq c*3^n$, meaning that $3 \leq c$.

The constants that make this statement true are $c \geq 3$ and $n_0 \geq 1$

Therefore 3^{n+1} is $O(3^n)$

Let us assume 3^{n+1} is $\Omega(3^n)$.

Therefore some constant c must exist such that $3^{n+1} \geq c * 3^n$.

This means that c must satisfy $3 \geq c$ for all n.

Since c can be any constant, we can say that 3^{n+1} is $\Omega(3^n)$

(b) 2^{2n} vs 2^n

If we wish to say 2^{2n} is $O(2^n)$, there must exist two constants c and n_0 such that $c * 2^n \geq 2^{2n}$ for all $n \geq n_0$

The constants that will fulfill this requirement are $c \geq 2$ and $n_0 \geq 1$

Therefore 2^{2n} is $O(2^n)$.

Let us assume that 2^{2n} is $\Omega(2^n)$.

Therefore some constant c must exist such that $2^{2n} \geq c * 2^n$.

This means that c must satisfy $2^n \geq c$ for all $n \geq n_0$

(c) 4^n vs 2^{2n}

Let us first simplify 2^{2n} to 4^n

For both $O(f(n))$ and $\Omega(f(n))$, all functions are big-O and Omega of themselves.

(d) n^2 vs $n^{2.01}$

Let us assume that n^2 is $O(n^{2.01})$.

That means that some constant c must exist such that $n^2 \leq c * n^{2.01}$

Therefore c must satisfy $1/n^{0.1} \leq c$ for all $n \geq n_0$

Since the function $1/n^{0.1}$ is a decreasing function we can say that n^2 is $O(n^{2.01})$.

Let us assume that n^2 is $\Omega(n^{2.01})$.

This means that some constant c must exist such that $n^2 \geq c * n^{2.01}$.

Therefore c must satisfy $1/n^0.1 \geq c$ for all $n \geq n_0$

Because $1/n^0.1$ is a decreasing function, there cannot exist a c that will satisfy this function.

Therefore n^2 is NOT $\Omega(n^{2.01})$.

(e) $n^{0.9}$ vs 0.9^n

Let us assume that $n^{0.9}$ is $O(0.9^n)$.

This means that some constant c must exist such that $n^{0.9} \leq c * 0.9^n$.

Therefore c must satisfy $e^{.9ln(n) - nln(.9)} \leq c$

This equation increases exponentially, and so there is no value of c that will satisfy this for all $n \geq n_0$.

Therefore $n^{0.9}$ is NOT $O(0.9^n)$.

For Ω the same is true except c must satisfy $e^{.9ln(n) - nln(.9)} \geq c$ instead.

We know that the equation is increasing at an exorbitant rate, so there definitely exists a c to satisfy this equation for all $n \geq n_0$.

Therefore $n^{0.9}$ is $\Omega(0.9^n)$.

2. Same instructions as problem 1.

(a) $\log^c n$ vs $\log n$, where c is a constant greater than 1.

Let us assume that $\log^c n$ is $O(\log n)$.

This means that some constant k must exist such that $log^c n \leq k * \log n$.

Therefore k must satisfy $\log^{c-1} n \leq k$.

$logn$ is an increasing function, so there exists no value of k that will satisfy this equation for all values of c or n.

For Ω, the inequality can be flipped to say that k must instead satisfy $\log^{c-1} n \leq k$.

Since $\log^{c-1} n$ is an increasing function, it is easy to find a value of k that will satisfy this equation for all values of c and n.

Therefore this set is only Ω.

(b) $\log n^c$ vs $\log n$, where $c = \Theta(1)$.

For this set of functions the constant k will again be used.

It must satisfy the inequality $c \leq k$ for O, and $c \geq k$ for Ω.

Since we can just set $k = c$, these are both satisfied for all values of c and n.

Therefore this set of functions is both O and Ω.

(c) $\log(c \cdot n)$ vs $\log n$, where $c = \Theta(1)$. You can assume both have the same base.

For this set of functions the const k will again be used.

Let us assume that $\log(c \cdot n)$ is $O(\log n)$.

This means that k must satisfy the equation $\log(c \cdot n) \leq k \cdot \log n$.

Which can be rewritten as $\log_n c - 1 \leq k$.

This function is decreasing, so we can say it is trivial to find a value of k that will satisfy all values of c and $n \geq n_0$.

We can flip the inequality for Ω, and by the same logic we can say that there exists no constant that will satisfy the flipped inequality.

Therefore this set is O and not Ω.

(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.

The restrictions exist because for logarithms with base < 1, any multiplication or division with it requires that the inequality be flipped.

For this equation, we can use c again!

Let us assume that $\log_a n$ is $O(\log_b n)$.

This means that c must satisfy the inequality $\log_a b \leq c$.

For Ω, the inequality is $\log_a b \geq c$.

Therefore we can say that c can equal $\log_a b$, satisfying both inequalities for all values of a, b, and n all at once.

This set is both O and Ω.

3. Let $f(x) = O(x)$ and $g(x) = O(x)$. Let c be a positive constant. Prove or disprove that $f(x) + c \cdot g(y) = O(x + y)$.

Let us assume that $f(x) + c \cdot g(y)$ is $O(x + y)$.

This means there exists some const k that satisfies the inequality

$f(x) + c \cdot g(y) \leq k \cdot (x + y)$

If $f(x) = O(x)$, then there exists some constant a that satisfies the inequality $f(x) \leq a \cdot x$.

Same for $g(x)$ and constant b.

If we say that k will satisfy
$k \geq a \cdot b \cdot c$, then
$k \cdot x \geq f(x)$ is true, and
$k \cdot y \geq c \cdot g(x)$ is true as well, therefore
$f(x) + c \cdot g(y) \leq k \cdot (x + y)$ is true as well.

Therefore $f(x) + c \cdot g(y) = O(x + y)$.

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.

For a geometric series, the nth partial sum is equal to $\frac{n(n+1)}{2}$.

Using this to simplify $f(n)$ to $\frac{\log^3 n \cdot n^{29} \cdot \log^3 n \cdot (n+1)^{29}}{2}$.

Simplify to $\frac{1}{2} \cdot \log^6 n \cdot n^{29} \cdot (n+1)^{29}$.

We can remove the constant in the beginning and the constant inside $(n+1)$ to further simplify.

$\log^6 n \cdot n^{58}$.

We proved earlier that $\log^c n = \Omega(\log n)$ and $n^c = \Omega(c^n)$.

Therefore we can further simplify to $\log n \cdot 58^n$. and 58^n is $\Omega(2^n)$, so

$g(n) = \log n \cdot 2^n$ and $f(n) = \Omega(g(n))$.

We had proved earlier that $\log^c n$ is not $O(\log n)$, so this does not work for big-O.