Caio Brighenti
COSC 302 - Analysis of Algorithms – Spring 2019
Lab 2

My assigned problem was #1. Problems 6,7, and 9 are my other three problems completed.

1. **Prove or disprove each of the following:**

   (a) $f(n) = O(g(n)) \implies g(n) = O(f(n))$

   We disprove by counterexample. Let $f(n) = n$ and $g(n) = n^2$. $n = O(n^2)$, but $n^2 \neq O(n)$. Thus, the claim is false.

   (b) $f(n) + g(n) = \Theta(min(f(n), g(n)))$

   We disprove by counterexample. Let $f(n) = n$ and $g(n) = n^2$. Then $n^2 + n \neq \Theta(n)$, therefore the claim is false.

   (c) $f(n) = O(g(n)) \implies log(f(n)) = O(log(g(n)))$

   We proceed by direct proof.

$$\begin{aligned} f(n) &\leq c_2 g(n) && \text{definition of O(g(n)} && (1)\\ log(f(n)) &\leq log(c_2 g(n)) && \text{log both sides} && (2)\\ &\leq log(c_2) + log(g(n)) && \text{log properties} && (3) \end{aligned}$$

   We now must find a constant d such that $log(f(n)) \leq dlog(g(n))$
   As we have that $f(n) \leq log(c_2) + log(g(n))$ we look for a d such that
   $log(c_2) + log(g(n)) \leq dlog(g(n))$.
   For this, since $g(n) \geq 1$, we can have $d = log(c) + 1$. Thus, we have:
   $f(n) \leq log(c_2) + log(g(n)) \leq dlog(g(n))$.

   As this is meets the definition of $log(f(n)) = O(log(g(n)))$, the claim must be true.

6. **Give definitions for $\Omega(g(n, m))$ and $\Theta(g(n, m))$**

   $\Omega(g(n, m)) = \{f(n, m)$: there exists positive constants c_2, n_0, and m_0
   such that $f(n, m) \leq c_2(g(n, m))$
   for all $n \geq n_0$ and $m \geq m_0\}$.

   $\Theta(g(n, m)) = \{f(n, m)$: there exists positive constants c_1, c_2, n_0, and m_0
   such that $c_1(g(n, m)) \leq f(n, m) \leq c_2(g(n, m))$
   for all $n \geq n_0$ and $m \geq m_0\}$.

7. **Is $2^{n+1} = O(2^n)$? Is $2^{2n} = O(2^n)$?**

   We have $2^{n+1} = 2 \cdot 2^n$. As $2 \cdot 2^n = \Theta(2^n)$, it follows that $2^{n+1} = \Theta(2^n)$.
   The second case is not true. We proceed by contradiction. Assume that $2^{2n} = O(2^n)$. We must thus have that:
   $2^{2n} \leq c_2 2^n$ by the definition of $\Theta(g(n))$. By rearranging, we have that:
   $\frac{2^{2n}}{2^n} \leq c_2$ and finally that:
   $2^n \leq c_2$.
   This statement is impossible, as a constant cannot be greater than or equal to a variable that grows to infinity. Thus, we have a contradiction, and the claim cannot be true.

6. **Let $f(n)$ and $g(n)$ be asymptotically non-negative functions. Using the basic definition of Θ-notation, prove that $max(f(n), g(n)) = \Theta(f(n) + g(n))$.**

As $f(n)$ and $g(n)$ are both asymptotically nonnegative, then for all $n > n_0$ it must be that $f(n) > 0$ and $g(n) > 0$. Thus, we must have that:
$f(n) + g(n) \geq f(n) \geq 0$ and that $f(n) + g(n) \geq g(n) \geq 0$.
Let $h(n) = max(f(n), g(n))$.
As $h(n)$ will always equal the biggest of the two functions, it is true that $h(n) \geq f(n)$ and $h(n) \geq g(n)$.
Thus, we can substitute $h(n)$ and obtain the following:

$$f(n) + g(n) \geq h(n) \geq 0$$

If we add a constant c_2, we can have that:

$$h(n) = max(f(n), g(n)) \leq c_2(f(n) + g(n))$$

Thus, we have $max(f(n), g(n)) = O(f(n) + g(n))$.
We also can write that $0 \leq f(n) \leq h(n)$ and $0 \leq g(n) \leq h(n)$. Adding these two inequalities results in:

$$0 \leq f(n) + g(n) \leq 2h(n)$$

Thus, by rearranging this we have:

$$h(n) = max(f(n), g(n)) \geq c_1(f(n) + g(n))$$

Thus, $max(f(n), g(n)) = \Omega(f(n) + g(n))$.
Since we have that $max(f(n), g(n)) = O(f(n) + g(n))$ and that $max(f(n), g(n)) = \Omega(f(n) + g(n))$, it must be that $max(f(n), g(n)) = \Theta(n) + g(n))$.