Algorithms (COT 6405): Solutions 4

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

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

Consider the following two functions:

$$\begin{aligned} f(n) &= 1 \\ g(n) &= \begin{cases} 1 & \text{if } n \text{ is even;} \\ n & \text{if } n \text{ is odd.} \end{cases} \end{aligned}$$

Since $f(n) \leq g(n)$, we immediately conclude that $f(n) = O(g(n))$. For even n, the function $f(n)$ is of the same order as $g(n)$, which means that $f(n) \neq o(g(n))$. On the other hand, for odd n, $f(n)$ grows asymptotically slower than $g(n)$, which implies that $f(n) \neq \Theta(g(n))$.

Problem 2
Prove the following transitivity property of asymptotic bounds:

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

Since $f(n) = \Theta(g(n))$, we conclude that there are some positive constants c_1, c_2, and n_1 such that, for all $n \geq n_1$, we have:

$$c_1 g(n) \leq f(n) \leq c_2 g(n).$$

Similarly, since $g(n) = \Theta(h(n))$, there are some positive constants c_3, c_4, and n_2 such that, for all $n \geq n_2$, we have:

$$c_3 h(n) \leq g(n) \leq c_4 h(n).$$

We combine these two inequalities as follows:

$$c_1 c_3 h(n) \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \leq c_2 c_4 h(n).$$

We now define three new constants, c_5, c_6, and n_3:

$c_5 = c_1 c_3$,
$c_6 = c_2 c_4$,
$n_3 = \max(n_1, n_2)$.

Then, the last inequality implies that, for every $n \geq n_3$, we have:

$$c_5 h(n) \leq f(n) \leq c_6 h(n).$$

This inequality means that, by definition, $f(n) = \Theta(h(n))$.