CS343: Complexity Proof Examples

Problem 1:

Prove that $2^n = O(2^{2n})$.

Proof: This is true if we can find a $c > 0$ and $n_0 > 0$ such that

$$0 \leq 2^n < c2^{2n}$$

for $n > n_0$.

Rewriting the RHS gives

$$0 \leq 2^n < c2^n 2^n$$

which clearly is true if

$$1 < c2^n$$

But this will be true if $c = 1$ and $n > 1$, or $n_0 = 1$.

Problem 2:

Prove that $max(f_1(n), f_2(n)) = \Omega(f_1(n) + f_2(n))$

Proof: This is true if we can find a $c > 0$ and $n_0 > 0$ such that

$$0 < c(f_1(n) + f_2(n)) \leq max(f_1(n), f_2(n))$$

for $n > n_0$.

Now, suppose for the moment that $max(f_1(n), f_2(n)) = f_1(n)$. This means that we must have $f_2(n) \leq f_1(n)$, so that

$$f_1(n) + f_2(n) \leq f_1(n) + f_1(n) = 2f_1(n) = 2max(f_1(n), f_2(n))$$

Dividing through by 2 and keeping only the first and last terms gives

$$(1/2)(f_1(n) + f_2(n)) \leq max(f_1(n), f_2(n))$$

Comparing this to what we want to prove, we find that $c = 1/2$ and $n_0 = 1$ will satisfy our original equation.