1. Let $f(n)$ and $g(n)$ be asymptotically nonnegative functions. Using the basic definitions of Θ-notation, prove that $\max(f(n), g(n)) = \Theta(f(n) + g(n))$.

For simplicity, let us define $h(n) = \max(f(n), g(n))$. To prove $h(n) = \Theta(f(n) + g(n))$, we must find positive constants c_1, c_2 and n_0 such that $0 \leq c_1 \cdot \{f(n) + g(n)\} \leq h(n) \leq c_2 \cdot \{f(n) + g(n)\}$ for all $n \geq n_0$.

Both $f(n)$ and $g(n)$ are asymptotically nonnegative functions, so that for some n' we have $0 \leq f(n)$ and $0 \leq g(n)$ for all $n \geq n'$. Select $n_0 = n'$.

Suppose further without loss of generality that $g(n) \leq f(n) = h(n)$. Then $f(n)+g(n) \leq 2 \cdot h(n)$, so $\frac{1}{2} \cdot (f(n) + g(n)) \leq h(n)$ and we can choose $c_1 = 1/2$. Likewise since both $f(n)$ and $g(n)$ are asymptotically nonnegative, we can write $f(n) \leq f(n) + g(n)$ for $n \geq n_0 = n'$ and choose $c_2 = 1$.

Hence with $c_1 = 1/2$, $c_2 = 1$ and $n_0 = n'$, we have $\max(f(n), g(n)) = \Theta(f(n) + g(n))$.

2. Show that for any real constants a and b, where $b > 0$,

$$(n + a)^b = \Theta(n^b)$$

To show that $(n + a)^b = \Theta(n^b)$, we must find positive constants c_1, c_2, and n_0 such that for all $n \geq n_0$, we have:

$$c_1 n^b \leq (n + a)^b \leq c_2 n^b$$

$$c_1 n^b \leq \{n(1 + \frac{a}{n})\}^b \leq c_2 n^b$$

$$c_1 n^b \leq n^b \left(1 + \frac{a}{n}\right)^b \leq c_2 n^b$$

$$c_1 \leq \left(1 + \frac{a}{n}\right)^b \leq c_2$$

Select $n_0 = 2|a|$, and suppose that $a < 0$. Then the middle term becomes $(1 + a/n_0)^b \to \left(\frac{1}{2}\right)^b$ when $n = n_0$ and increases as $n \geq n_0$, which suggests that we should choose $c_1 = \left(\frac{1}{2}\right)^b$. Likewise, suppose instead that we have $a > 0$. Then the middle term becomes $(1 + a/n_0)^b \to \left(\frac{3}{2}\right)^b$ when $n = n_0$ and decreases as $n \geq n_0$, which suggests that we should choose $c_2 = \left(\frac{3}{2}\right)^b$.

Hence with $c_1 = \left(\frac{1}{2}\right)^b$, $c_2 = \left(\frac{3}{2}\right)^b$ and $n_0 = 2|a|$, we have $(n+a)^b = \Theta(n^b)$.

3. Explain why the statement "The running time of algorithm A is at least $O(n^2)$" is content-free.

The statement doesn't make sense, or is meaningless, because $O(n^2)$ is an *upper* bound on the running time of A. It could make sense to that the running time of A is at *most* $O(n^2)$, or it could also make sense to say that the running time of A is at *best* $\Omega(n^2)$, but the statement as it stands doesn't make sense in the context of what $O(n^2)$ means.

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

To determine if $2^{n+1} = O(2^n)$, we must find positive constants c and n_0 such that $2^{n+1} \leq c \cdot 2^n$ for all $n \geq n_0$. Dividing both sides of this inequality by 2^n, we obtain $2 \leq c$. Select $n_0 = 1$ and $c = 3$, and we have established that $2^{n+1} = O(2^n)$.

To determine if $2^{2n} = O(2^n)$, we must find positive constants c and n_0 such that $2^{2n} \leq c \cdot 2^n$ for all $n \geq n_0$. Dividing both sides of this inequality by 2^n, we obtain $2^n \leq c$. Clearly, there is no choice of c that will satisfy this relation. Hence, we conclude that $2^{2n} \neq O(2^n)$.

5. Prove Theorem 2.1.

Theorem 2.1 states that for any two functions $f(n)$ and $g(n)$, $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

Suppose that $f(n) = \Theta(g(n))$. This means that we can find positive constants c_1, c_2 and n_0 such that $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n \geq n_0$. Clearly, this means that we have found positive constants c_2 and n_0 such that $0 \leq f(n) \leq c_2 g(n)$ for all $n \geq n_0$, hence $f(n) = O(g(n))$. Likewise, it also means that we have found positive constants c_1 and n_0 such that $0 \leq c_1 g(n) \leq f(n)$ for all $n \geq n_0$, hence $f(n) = \Omega(g(n))$.

This establishes $f(n) = \Theta(g(n)) \Rightarrow f(n) = O(g(n)) \wedge f(n) = \Omega(g(n))$.

Suppose next that $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$. Since $f(n) = \Omega(g(n))$, we can find positive constants c_1 and n_1 such that $0 \leq c_1 g(n) \leq f(n)$ for all $n \geq n_1$. Likewise, since $f(n) = O(g(n))$, we can find positive constants c_2 and n_2 such that $0 \leq f(n) \leq c_2 g(n)$ for all $n \geq n_2$. Let $n_0 = \max(n_1, n_2)$. We have now identified positive constants c_1, c_2 and n_0 such that $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n \geq n_0$. Hence, $f(n) = \Theta(g(n))$.

This establishes $f(n) = O(g(n)) \wedge f(n) = \Omega(g(n)) \Rightarrow f(n) = \Theta(g(n))$.

Hence $f(n) = \Theta(g(n)) \iff f(n) = O(g(n)) \wedge f(n) = \Omega(g(n))$.

6. Prove that the running time of an algorithm is $\Theta(g(n))$ if and only if its worst-case running time is $O(g(n))$ and its best-case running time is $\Omega(g(n))$.

Let $T(n)$ be the running time of the algorithm and suppose that $T(n) = \Theta(g(n))$. Hence we can find positive numbers c_1, c_2 and n_0 such that $0 \leq c_1 \cdot g(n) \leq T(n) \leq c_2 \cdot g(n)$ for all $n \geq n_0$. The longest $T(n)$ could possibly take to complete is bounded above by $c_2 \cdot g(n)$, hence $T(n) = O(g(n))$. The fastest that $T(n)$ could possibly take to complete is bounded below by $c_1 \cdot g(n)$, hence $T(n) = \Omega(g(n))$.

Suppose that the fastest running time for $T(n)$ is bounded below by $\Omega(g(n))$. This means that we can find positive constants c_1 and n_1 such that $0 \leq c_1 \cdot g(n) \leq f(n)$ for all $n \geq n_1$. Suppose further that the worst-case running time for $T(n)$ is bounded above by $O(g(n))$. This means that we can find positive constants c_2 and n_2 such that $0 \leq f(n) \leq c_2 \cdot g(n)$ for all $n \geq n_2$. Let $n_0 = \max(n_1, n_2)$. Therefore, we have $0 \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$ for all $n \geq n_0$ and hence $T(n) = \Theta(g(n))$.

7. Prove that $o(g(n)) \cap \omega(g(n))$ is the empty set.

The notations $o(g(n))$ and $\omega(g(n))$ are used to indicate bounds that are *not* asymptotically tight. That is, we have $f(n) = o(g(n))$ if for *any* positive constant c, we can find a positive constant n_0 such that $0 \leq f(n) < cg(n)$ for all $n \geq n_0$. Likewise, we have $f(n) = \omega(g(n))$ if for *any* positive constant c, we can find a positive constant n_0 such that $0 \leq cg(n) < f(n)$ for all $n \geq n_0$.

Suppose we have $f(n) = \omega(g(n))$, and select an arbitrary positive constant $c_0 > 0$. Since $f(n) = \omega(g(n))$, we can find a positive constant $n_1 > 0$ such that $0 \leq c_0 \cdot g(n) < f(n)$ for all $n \geq n_1$. Suppose further that $f(n) = o(g(n))$, so that we can find another positive constant $n_2 > 0$ such that $0 \leq f(n) < c_0 \cdot g(n)$ for all $n \geq n_2$. Let $n_0 = \max(n_1, n_2)$. It follows that we must have $c_0 \cdot g(n) < f(n) < c_0 \cdot g(n)$ for all $n \geq n_0$. This is a contradiction, and so $f(n)$ cannot be in both $\omega(g(n))$ and $o(g(n))$ simultaneously. Since the choice of $f(n)$ was arbitrary, we have $o(g(n)) \cap \omega(g(n)) = \emptyset$.

A second approach would be to use a result given in the text, namely that $f(n) = \omega(g(n))$ if and only if $g(n) = o(f(n))$. Suppose that $\omega(g(n)) \cap o(g(n)) \neq \emptyset$ and choose some $f(n) \in \omega(g(n)) \cap o(g(n))$. Then $f(n) = \omega(g(n))$ and $f(n) = o(g(n))$. But since $f(n) = \omega(g(n))$, we must have that $g(n) = o(f(n))$ from the result in the text.

Since $f(n) = o(g(n))$, we can find a $n_1 > 0$ such that $0 \leq f(n) < c \cdot g(n)$ for any $c > 0$ and all $n \geq n_1$. Likewise, since $g(n) = o(f(n))$, we can find $n_2 > 0$ such that $0 \leq g(n) < c \cdot f(n)$ for any $c > 0$ and all $n \geq n_2$. Choose $n_0 = \max(n_1, n_2)$. It follows then that we have $f(n) < c \cdot g(n)$ and $g(n) < c \cdot f(n)$ for any $c > 0$ and all $n \geq n_0$, which is a contradiction. Hence, no such $f(n)$ exists and $o(g(n)) \cap \omega(g(n)) = \emptyset$.

8. We can extend our notation to the case of two parameters n and m that can go to infinity independently at different rates. For a given function $g(n, m)$, we denote by $O(g(n, m))$ the set of functions

$O(g(n, m)) = \{f(n, m)$: there exist positive constants c, n_0 and m_0 such that $0 \leq f(n, m) \leq c \cdot g(n, m)$ for all $n \geq n_0$ and $m \geq m_0\}$

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

$\Omega(g(n, m)) = \{f(n, m)$: there exist positive constants c, n_0 and m_0 such that $0 \leq c \cdot g(n, m) \leq f(n, m)$ for all $n \geq n_0$ and $m \geq m_0$ $\}$.

$\Theta(g(n, m)) = \{f(n, m)$: there exist positive constants c_1, c_2, n_0 and m_0 such that $0 \leq c_1 \cdot g(n, m) \leq f(n, m) \leq c_2 \cdot g(n, m)$ for all $n \geq n_0$ and $m \geq m_0\}$.