2. State whether the following assertions are true or false. If any statements are false, give a related statement which is true.

a. $f(n) = O(g(n))$ implies $f(n) = o(g(n))$.  **False**
   Related True Statement: $f(n) = o(g(n))$ implies $f(n) = O(g(n))$

b. $f(n) = O(g(n))$ if and only if $g(n) = \Omega(f(n))$.  **True**

c. $f(n) = \Theta(g(n))$ if and only if $\lim_{n \to \infty} (f(n)/g(n)) = L$, where $0 < L < \infty$.  **False**
   Related True Statement: $0 < L < \infty$ and $\lim_{n \to \infty} (f(n)/g(n)) = L$ implies $f(n) = \Theta(g(n))$

3. Prove that $\Theta(f(n)) \cdot \Theta(g(n)) = \Theta(f(n) \cdot g(n))$. In other words, if $h_1(n) = \Theta(f(n))$ and $h_2(n) = \Theta(g(n))$, then $h_1(n) \cdot h_2(n) = \Theta(f(n) \cdot g(n))$.

**Proof:**
By hypothesis there exist positive constants $n_1$, $n_2$, $a_1$, $b_1$, $a_2$, and $b_2$ such that

\[
\forall n \geq n_1: \quad 0 \leq a_1 f(n) \leq h_1(n) \leq b_1 f(n)
\]

and

\[
\forall n \geq n_2: \quad 0 \leq a_2 g(n) \leq h_2(n) \leq b_2 g(n)
\]

If $n \geq n_0 = \max(n_1, n_2)$, then both inequalities hold. Let $c = a_1 a_2$, and $d = b_1 b_2$. Since everything in sight is non-negative, we can multiply these two inequalities to get

\[
\forall n \geq n_0: 0 \leq c f(n) g(n) \leq h_1(n) h_2(n) \leq d f(n) g(n),
\]

and hence $h_1(n) \cdot h_2(n) = \Theta(f(n) \cdot g(n))$ as required.  ■

4. Let $f(n)$ and $g(n)$ be asymptotically positive functions (i.e. $f(n) > 0$ and $g(n) > 0$ for all sufficiently large $n$), and suppose that $f(n) = \Theta(g(n))$. Does it necessarily follow that $\frac{1}{f(n)} = \Theta\left(\frac{1}{g(n)}\right)$? Either prove this statement, or give a counter-example.

**Solution:**
The statement is true, as we now prove. By hypothesis there exist positive numbers $c_1$, $c_2$, and $n_0$ such that for all $n \geq n_0$: $0 < c_1 g(n) \leq f(n) \leq c_2 g(n)$. (Note: the strict inequality $<$ on the left follows from the fact that $f(n)$ and $g(n)$ are asymptotically positive.) Taking the reciprocals of all the positive terms in this inequality gives: $0 < \frac{1}{c_2} \cdot \frac{1}{g(n)} \leq \frac{1}{f(n)} \leq \frac{1}{c_1} \cdot \frac{1}{g(n)}$ for all $n \geq n_0$. Observe that both $\frac{1}{c_2} > 0$ and $\frac{1}{c_1} > 0$, whence $\frac{1}{f(n)} = \Theta\left(\frac{1}{g(n)}\right)$.  ■
5. Give an example of functions $f(n)$ and $g(n)$ such that $f(n) = o(g(n))$ but $\log(f(n)) \neq o(\log(g(n)))$. (Hint: Consider $n!$ and $n^n$ and use the corollary to Stirling’s formula in the handout on common functions.)

Solution:
Following the hint, we let $f(n) = n!$ and $g(n) = n^n$. Part (1) of the Corollary to Stirling’s formula on page 3 of the handout on common functions showed that $f(n) = o(g(n))$. Part (3) of that same Corollary showed that $\log(n!) = \Theta(n \log(n))$. Hence $\log(f(n)) = \Theta(n \log(n)) = \Theta(\log(n^n)) = \Theta(\log(g(n)))$. Since $\Theta(\log(g(n))) \cap \Theta(\log(g(n))) = \emptyset$, we have $\log(f(n)) \neq o(\log(g(n)))$.

7. (d) Use limits to prove the following: $f(n) + o(f(n)) = \Theta(f(n))$

Proof:
In this equation, the term $o(f(n))$ stands for some function $h(n)$ satisfying $\lim_{n \to \infty} \left( \frac{h(n)}{f(n)} \right) = 0$. Therefore $\lim_{n \to \infty} \left( \frac{f(n) + h(n)}{f(n)} \right) = \lim_{n \to \infty} \left( 1 + \frac{h(n)}{f(n)} \right) = 1 + \lim_{n \to \infty} \left( \frac{h(n)}{f(n)} \right) = 1$, showing that $f(n) + h(n) = \Theta(f(n))$. Note that this result justifies the practice of dropping low order terms when finding the asymptotic growth rate of a function.

8. Let $g(n) = n$ and $f(n) = n + \frac{1}{2} n^2 (\sin(n) + 1)$. Show that
   a. $f(n) = \Omega(g(n))$
   b. $f(n) \neq O(g(n))$
   c. $\lim_{n \to \infty} \left( \frac{f(n)}{g(n)} \right)$ does not exist, even in the sense of being infinite.

Note: this is the ‘Example C’ mentioned in the handout on asymptotic growth rates.

Proof of (a):
For any $n \geq 1$ we have $-1 \leq \sin(n) \leq 1$ and hence $\sin(n) + 1 \geq 0$. Thus

$$f(n) = n + \frac{1}{2} n^2 (\sin(n) + 1) \geq n = g(n).$$

Thus $0 \leq 1 \cdot g(n) \leq f(n)$ for all $n \geq 1$, whence $f(n) = \Omega(g(n))$.

Proof of (b):
We must show that the sentence ‘$\exists c > 0, \exists n_0 > 0, \forall n \geq n_0 : 0 \leq f(n) \leq c \cdot g(n)$’ is false. We do this by showing that it’s negation ‘$\forall c > 0, \forall n_0 > 0, \exists n \geq n_0 : c \cdot g(n) < f(n)$’ is true. Pick $c > 0$ and $n_0 > 0$ arbitrarily. Define $n = \frac{\pi}{2} + 2\pi \cdot k$ where the integer $k$ is chosen so large as to guarantee that $n \geq \max(c, n_0)$. (This is possible since $\frac{\pi}{2} + 2\pi \cdot k \to \infty$ as $k \to \infty$.) Then $n \geq n_0$ and $n \geq c > c - 1$, whence $n + 1 > c$. Observe also that $\sin(n) = 1$, and therefore

$$f(n) = n + \frac{1}{2} n^2 (\sin(n) + 1) = n + n^2 = n(1 + n) > cn = c \cdot g(n)$$

as required.

Proof of (c):
Observe that

$$\frac{f(n)}{g(n)} \leq \frac{n + \frac{1}{2} n^2 (\sin(n) + 1)}{n} = 1 + \frac{1}{2} n (\sin(n) + 1),$$

which oscillates with increasing amplitude between 1 and $1 + n$ as $n \to \infty$, and therefore has no limit, even in the sense of being infinite.
11. Consider the following *sketch* of an algorithm called 
ProcessArray that performs some unspecified 
operation on a subarray \( A[p \ldots r] \).

\[
\text{ProcessArray}(A, p, r) \quad \text{(Preconditions: } 1 \leq p \text{ and } r \leq \text{length}[A])
\]

1. Perform 1 basic operation
2. if \( p < r \)
   \[ q = \left\lfloor \frac{p+r}{2} \right\rfloor \]
3. ProcessArray(A, p, q)
4. ProcessArray(A, q+1, r)

a. Write a recurrence formula for the number \( T(n) \) of basic operations performed by this algorithm when 
called on the full array \( A[1 \ldots n] \), i.e. by 
ProcessArray(A, 1, n). (Hint: recall our analysis of 
MergeSort.)

**Solution:**

\[
T(n) = \begin{cases} 
1 & n = 1 \\
T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1 & n \geq 2 
\end{cases}
\]

b. Show that the solution to this recurrence is \( T(n) = 2n - 1 \), whence \( T(n) = \Theta(n) \).

**Proof:**

Observe that when \( n = 1 \) we have \( T(1) = 2 \cdot 1 - 1 = 1 \). When \( n \geq 2 \) we have

\[
\text{RHS} = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1 \\
= (2\lfloor n/2 \rfloor - 1) + (2\lceil n/2 \rceil - 1) + 1 \\
= 2(\lfloor n/2 \rfloor + \lceil n/2 \rceil) - 1 \\
= 2n - 1 \\
= T(n) \\
= \text{LHS}
\]

12. Consider the following algorithm that does nothing but waste time:

\[
\text{WasteTime}(n) \quad \text{ (pre: } n \geq 1) 
\]

1. if \( n > 1 \)
2. for \( i = 1 \) to \( n^3 \)
3. waste 2 units of time
4. for \( i = 1 \) to 7
5. WasteTime(\( \lceil n/2 \rceil \))
6. waste 3 units of time

a. Write a recurrence formula for the amount of time \( T(n) \) wasted by this algorithm.

**Solution:**

\[
T(n) = \begin{cases} 
0 & n = 1 \\
7T(\lfloor n/2 \rfloor) + 2n^3 + 3 & n \geq 2 
\end{cases}
\]

b. Show that when \( n \) is an exact power of 2, the explicit solution to this recurrence relation is given by

\( T(n) = 16n^3 - \frac{1}{2} - \frac{31}{2}n \lg 7 \), and hence \( T(n) = \Theta(n^3) \).
Proof:
If \( n = 1 \) then \( T(1) = 16 \cdot 1^3 - \frac{1}{2} - \frac{31}{2} \cdot 1 \cdot 7 = 16 - \frac{32}{2} = 0 \). When \( n \geq 2 \) is an exact power of 2 we have

\[
\text{RHS} = 7T(n/2) + 2n^3 + 3
\]

\[
= 7 \left( 16 \left( \frac{n}{2} \right)^3 - \frac{1}{2} - \frac{31}{2} \left( \frac{n}{2} \right)^{\lg 7} \right) + 2n^3 + 3
\]

\[
= 7 \left( \frac{16}{8} n^3 - \frac{1}{2} - \frac{31}{2} \left( \frac{n^{\lg 7}}{7} \right) \right) + 2n^3 + 3
\]

\[
= 14n^3 - \frac{7}{2} - \frac{31}{2} n^{\lg 7} + 2n^3 + \frac{6}{2}
\]

\[
= 16n^3 - \frac{1}{2} - \frac{31}{2} n^{\lg 7}
\]

\[
= T(n)
\]

\[
= \text{LHS}
\]

13. Define \( T(n) \) by the recurrence formula

\[
T(n) = \begin{cases} 
1 & 1 \leq n < 3 \\
2T([n/3]) + 4n & n \geq 3
\end{cases}
\]

Use Induction to show that \( \forall n \geq 1: T(n) \leq 12n \), and hence \( T(n) = O(n) \).

Proof: (Multiple base cases, strong version)
I. Observe \( T(1) = 1 \leq 12 \cdot 1 \) and \( T(2) = 1 \leq 12 \cdot 2 \), so the base cases are satisfied.
IId. Let \( n \geq 3 \) and assume for all \( k \) in the range \( 1 \leq k < n \) that \( T(k) \leq 12k \). Since \( 1 \leq [n/3] \leq n \), we have \( T([n/3]) \leq 12[n/3] \). We must show that \( T(n) \leq 12n \). Observe

\[
T(n) = 2T([n/3]) + 4n \quad \text{by the recurrence formula for } T(n)
\]

\[
\leq 2 \cdot 12[n/3] + 4n \quad \text{by the induction hypothesis}
\]

\[
\leq 2 \cdot 12(n/3) + 4n \quad \text{since } [x] \leq x \text{ for any real number } x
\]

\[
= 8n + 4n
\]

\[
= 12n
\]

The result now holds for all \( n \geq 3 \).
15. Define $S(n)$ for $n \in \mathbb{Z}^+$ by the recurrence:

$$S(n) = \begin{cases} 
0 & \text{if } n = 1 \\
S([n/2]) + 1 & \text{if } n \geq 2 
\end{cases}$$

Use induction to prove that $S(n) \geq \lg(n)$ for all $n \geq 1$, and hence $S(n) = \Omega(\lg n)$.

**Proof:** Let $P(n)$ be the inequality $S(n) \geq \lg(n)$.

I. The inequality $S(1) \geq \lg(1)$ reduces to $0 \geq 0$, which is obviously true, so $P(1)$ holds.

II. Let $n > 1$ and assume for all $k$ in the range $1 \leq k < n$ that $S(k) \geq \lg(k)$. Then

$$S(n) = S([n/2]) + 1 \quad \text{by the definition of } S(n)$$
$$\geq \lg[n/2] + 1 \quad \text{by the induction hypothesis with } k = [n/2]$$
$$\geq \lg(n/2) + 1 \quad \text{since } [x] \geq x \text{ for any } x$$
$$= \lg(n) - \lg(2) + 1$$
$$= \lg(n)$$

showing that $P(n)$ holds. Therefore $S(n) \geq \lg(n)$ for all $n \geq 1$, as claimed.

16. Let $f(n)$ be a positive, increasing function that satisfies $f(n/2) = \Theta(f(n))$. Show that

$$\sum_{i=1}^{n} f(i) = \Theta(nf(n))$$

(Hint: Emulate the Example on page 4 of the handout on asymptotic growth rates in which it is proved that $\sum_{i=1}^{n} t^k = \Theta(n^{k+1})$ for any positive integer $k$.)

**Proof:**

Since $f(n)$ is increasing we have $\sum_{i=1}^{n} f(i) \leq \sum_{i=1}^{n} f(n) = nf(n) = O(nf(n))$. Note also that

$$\sum_{i=1}^{n} f(i) \geq \sum_{i=[n/2]}^{n} f(i) \quad \text{by discarding some positive terms}$$
$$\geq \sum_{i=[n/2]}^{n} f([n/2]) \quad \text{since } f(n) \text{ is increasing}$$
$$= (n - [n/2] + 1)f([n/2]) \quad \text{by counting terms}$$
$$= ([n/2] + 1)f([n/2]) \quad \text{since } n = [n/2] + [n/2]$$
$$> ((n/2) - 1 + 1)f(n/2) \quad \text{since } f(n) \text{ is increasing, } [x] \geq x, \text{ and } [x] > x - 1$$
$$= (n/2)f(n/2)$$
$$= \Omega(nf(n)) \quad \text{since } f(n/2) = \Theta(f(n)), \text{ whence } f(n/2) = \Omega(f(n))$$

It follows from an exercise in the handout on Asymptotic Growth rates that $\sum_{i=1}^{n} f(i) = \Theta(nf(n))$, as claimed.
17. Use the result of the preceding problem to give an alternate proof of \( \log(n!) = \Theta(n \log(n)) \) that does not use Stirling’s formula.

**Proof:**
Observe that \( \log(n) \) is a positive, increasing function, and \( \log(n/2) = \log(n) - \log(2) = \Theta(\log(n)) \). We may therefore apply the result of the preceding problem with \( f(n) = \log(n) \), and properties of logarithms to get

\[
\log(n!) = \sum_{i=1}^{n} \log(i) = \Theta(n \log(n))
\]

as claimed.

18. Let \( T(n) \) be defined by the recurrence formula

\[
T(n) = \begin{cases} 
1 & \text{if } n = 1 \\
T(\lfloor n/2 \rfloor) + n^2 & \text{if } n \geq 2 
\end{cases}
\]

Show that \( \forall n \geq 1: T(n) \leq \frac{4}{3} n^2 \), and hence \( T(n) = O(n^2) \).

**Proof:**
Let \( P(n) \) be the statement \( T(n) \leq (4/3)n^2 \). Then \( P(1) \) is true, since \( T(1) = 1 \leq 4/3 = (4/3) \cdot 1^2 \), and the base case is satisfied.

Let \( n > 1 \) be chosen arbitrarily, and suppose for all \( k \) in the range \( 1 \leq k < n \) that \( T(k) \leq (4/3)k^2 \). We must show as a consequence that \( T(n) \leq (4/3)n^2 \). Observe

\[
T(n) = T(\lfloor n/2 \rfloor) + n^2 \quad \text{by the recurrence formula for } T(n)
\leq (4/3)\lfloor n/2 \rfloor^2 + n^2 \quad \text{by the induction hypothesis with } k = \lfloor n/2 \rfloor
\leq (4/3)(n/2)^2 + n^2 \quad \text{since } \lfloor x \rfloor \leq x \text{ for any } x
= n^2/3 + n^2
= (4/3)n^2,
\]

as required.