The following are examples of answers to (hypothetical) homework questions on asymptotics that would receive full marks.

1. Show, using the definition of $\Theta$, that $n + \sqrt{n} = \Theta(n)$.

Taking $c = 2$ and $N = 1$, $n + \sqrt{n}$ is at most $2n$ for all $n \geq N$, so $n + \sqrt{n}$ is in $O(n)$. Taking $c = 1$ and $N = 1$, $n + \sqrt{n}$ is at least $n$ for all $n \geq N$, so $n + \sqrt{n}$ is in $\Omega(n)$. Therefore, $n + \sqrt{n}$ is in $\Theta(n)$.

2. Find the order of growth (up to $\Theta$) of $\sum_{i=0}^{n-1} i^3$, showing your work.

$$\Theta\left(\sum_{i=0}^{n-1} i^3\right) = \Theta\left(\int_0^n x^3 dx\right)$$

$$= \Theta\left(\frac{x^4}{4}\right)\bigg|_0^n$$

$$= \Theta\left(\frac{n^4}{4}\right)$$

$$= \Theta(n^4)$$

3. Prove that, for every $a$ and $b$ greater than one, $\log_a n = \Theta(\log_b n)$.

Using L'Hôpital's Rule, the limit of the ratio of the two functions as $n$ goes to infinity is a constant:

$$\lim_{n \to \infty} \frac{\log_a n}{\log_b n} = \lim_{n \to \infty} \frac{1/(n \ln a)}{1/(n \ln b)}$$

$$= \lim_{n \to \infty} \frac{\ln b}{\ln a}$$

$$= \frac{\ln b}{\ln a}$$

And $\ln b$ and $\ln a$ are both positive, so the constant $\frac{\ln b}{\ln a}$ is also positive. Therefore, by the limit method, $\log_a n = \Theta(\log_b n)$. 
4. Consider the following pseudocode:

```
Input: A, a set of numbers
1 let r ← |A|
2 for a ∈ A do
3     let r ← r + a
4 end
5 for a ∈ A do
6     let r ← r · a
7 end
8 return r
```

a. What is \( n \), the input size for this pseudocode?

There is only one input, and its size affects the algorithm's runtime, so \( n \) is the cardinality of \( A \).

b. For the purposes of analyzing time complexity, what are the basic operation(s) in this pseudocode?

The algorithm uses four operations that could qualify: taking the cardinality of a set, advancing to the next set element as part of a for-each loop, adding two numbers, and multiplying two numbers. To make sure that both loops are represented, we can either choose to have one basic operation that is advancing to the next set element or we can choose to have two basic operations, addition and multiplication. The additions and multiplications are easier to see and reason about, so we will choose them.

c. What is this algorithm's asymptotic time complexity? Show your work.

A single run of line 3 involves one addition, taking 1 step. The loop on lines 2 through 4 runs for \(|A|\) iterations, so it therefore takes \( \sum_{i=0}^{n-1} 1 \) steps in total. (Here, we have \( i \) ranging over the indices that \( A \) would have if we put it into iteration order.)

Similarly, a single run of line 6 involves one multiplication, taking 1 step, so the loop on lines 5 through 7, which also runs for \(|A|\) iterations, takes \( \sum_{j=0}^{n-1} 1 \) steps in total. In total, the algorithm needs

\[
\left( \sum_{i=0}^{n-1} 1 \right) + \left( \sum_{j=0}^{n-1} 1 \right) = (i|_0^n) + (j|_0^n)
\]

\[= n + n = \Theta(n)\]

steps to run.

d. Where are the bottleneck(s)? I.e., which loop(s) dominate the runtime for large \( n \)?

The first loops runs in \( n = \Theta(n) \) time, which matches the asymptotic runtime of the whole algorithm, so it is a bottleneck. But the second loop also runs in \( n = \Theta(n) \) time, so it too is a bottleneck. In other words, we would have to improve both loops before we could see any improvement in the whole algorithm's scalability.
5. Part of the application you are working on needs to find a number that is in both of two lists, $A$ and $B$. Prove by contradiction that every algorithm that solves this problem takes $\Omega(\min(|A|, |B|))$ time in the worst case.

Suppose by contradiction that some algorithm runs in better than $\Omega(\min(|A|, |B|))$ time in its worst case. Then, by the definition of $\Omega$, it solves the problem for large enough inputs with fewer than $\min(|A|, |B|)$ list accesses, and by the definition of “worst case”, this is true regardless of what we choose for the list elements.

So build an input as follows: run the algorithm, and whenever it accesses an element of $A$ that it has not accessed before, put a new negative number there, and whenever it accesses an element of $B$ that it has not accessed before, put a new positive number there. By the pigeonhole principle, there will still be unaccessed elements in both lists when the algorithm terminates, so whatever answer the algorithm gives, choose a different number to fill those undetermined spots. None of the negatives can match any of the positives, so the only right answer is the number we chose at the end, which is not the answer the algorithm gave, contradicting its correctness.
6. Consider the problem of packing five $3 \times 1$ parcels into a $4 \times 4$ box. Let $P$ be a predicate such that $P(s)$ holds if and only if there is a packing that leaves the $1 \times 1$ square $s$ empty. Prove that $|\{s \mid P(s)\}| = 4$.

A $3 \times 4$ block of parcels with the fifth parcel alongside leaves a corner empty, so, by symmetry, any corner square satisfies $P$, and $|\{s \mid P(s)\}|$ is at least four. Suppose the count is actually greater than four, which can only happen if some packing fills every corner. But any parcel that fills a corner must lie along a side of the box, so four of them would leave only the $2 \times 2$ interior for the fifth parcel. It cannot fit there—a contradiction.

7. Prove by induction that $2^n = 1 + \sum_{i=0}^{n-1} 2^i$ for all naturals $n$.

An empty sum is zero, so the $n = 0$ case holds trivially. For any larger $n$, we can suppose by induction that $2^{n-1} = 1 + \sum_{i=0}^{n-2} 2^i$, and adding $2^{n-1}$ to both sides establishes the claim.

8. A tree $T'$ is formed by adding a single vertex $v$ and some number of edges to the nonempty tree $T$. Prove by contradiction that $v$ has degree one in $T'$.

If $v$ has degree zero, it is disconnected from $T$'s vertices in $T'$, contradicting the fact that $T'$ is a tree. On the other hand, if $v$ has degree at least two, then between any two of its neighbors $x$ and $y$ there is a path $[x, \ldots, y]$ through $T$ that necessarily avoids $v$, which would make $[x, \ldots, y, v]$ a cycle, also contradicting the fact that $T'$ is a tree.

9. Let $Q$ be a predicate on graphs such that $Q(G)$ holds if and only if it is possible to color the vertices of $G$ with the colors red, green, and blue such that every vertex is not adjacent to any vertices of its own color, but is adjacent to at least one vertex of each of the other two colors. Prove by induction that for every $n$ greater than two there is a $Q$-satisfying graph of size $n$.

We induct on $n$. A complete graph on three vertices can be colored with all three colors, establishing a base case at $n = 3$. For larger $n$, the inductive hypothesis ensures that there is a graph $G$ of size $n - 1$ and a coloring $C$ by which $G$ satisfies $Q$. Choose any one vertex in $G$, create a copy of it with the same neighborhood, and then color the new graph according to $C$, using the original vertex’s color for its copy. By construction, the new graph satisfies $Q$, so the result follows by the principle of induction.