CS2205: Theory of Computation
Problem Class XIII

$t(n)$ is $O(f(n))$ if there are positive constants n_0 and c such that, for all $n \geq n_0$, $t(n) \leq c \times f(n)$

1. Prove the following:
   (a) $3n^3 + 4n + 6$ is $O(n^3)$
   (b) $3n^5 + 10n^4 - 4n^3 + n + 1$ is $O(n^5)$
   (c) 17 is $O(1)$
   (d) $\frac{n(n-1)}{2}$ is $O(n^2)$
   (e) $2n^2 + 3n + 4$ is $O(0.001n^3)$
   (f) $2^n + n$ is $O(2^n)$
   (g) $5n + 2^{n+10} + 3$ is $O(2^n)$
   (h) $2n + \log n + 6$ is $O(n)$
   (i) $20n^3 + 10n \log n + 5$ is $O(n^3)$
   (j) $3 \log n + \log(\log n)$ is $O(\log n)$

2. Prove the following:
   (a) n^3 is not $O(n^2)$
   (b) 3^n is not $O(2^n)$.

3. Decide whether or not the following hold. In each case, prove what you decide.
   (a) n^2 is $O(n^2 + 100n)$
   (b) $n^2 + 100n$ is $O(n^2)$
   (c) 2^{2n} is $O(2^n)$

4. Write out the following list of functions ordered by growth rate, from lowest to highest. If you think any of them have the same growth rate, circle them together in your list.

$n \log(\log n)$	37	$n^{1.5}$	n^3
$\sqrt{n}$	$\frac{2}{n}$	$n \log n$	2^n
$n \log(n^2)$	n	$n^2 \log n$	n^2

5. Dr. Toc, the brilliant but unreliable theorist, has been wreaking havoc again.
   (a) Dr. Toc presents an algorithm at a conference and announces that its worst-case time complexity is $O(\log n)$. A member of the audience asks whether the good doctor is taking logarithms to the base 2. Dr. Toc replies, "Idiot! In big-Oh relationships, the base of the logarithms doesn't matter." Is the good doctor right? Explain your answer in detail. (Hint. Recall the following identity: $\log_b n = \frac{\log_c n}{\log_c b}$.)
   (b) At the same conference, Dr. Toc presents another algorithm. The good doctor tells the audience, "The worst-case time complexity of my new algorithm is *at least* $O(n^2)$." Explain why the phrase "at least" is redundant.