Assignment #1:

1. Let P be a problem. The worst-case time complexity of P is $O(n^2)$. The worst case time complexity of P is also $\Omega(n \lg n)$. Let A be an algorithm that solves P. Which subset of the following statements are consistent with this information about the complexity of P? Briefly explain your answer.

    (a) A has worst-case time complexity $O(n^{\frac{3}{2}})$.
    No statement about any particular alogorithm can contradict the statement "The worst-case time complexity of P is $O(n^2)$". Consistent: The worst case complexity of P being $\Omega(n \lg n)$ does not prevent it from being $O(n^{\frac{3}{2}})$ since $n^{\frac{3}{2}} = \Omega(n \lg n)$.

    (b) A has worst-case time complexity $O(n)$.
    NO since P has worst case complexity $\Omega(n \lg n)$ and $n = o(n \lg n)$ and hence $n \neq \Omega(n \lg n)$

    (c) A has worst-case time complexity $\Theta(n^2)$.
    Consistent since $\Theta(n^2)$ does not contradict $\Omega(n \lg n)$.

2. For each of these questions, briefly explain your answer.

    (a) If I prove that an algorithm takes $O(n^2)$ worst-case time, is it possible that it takes $O(n)$ on some inputs?
    Yes; these inputs may be nicer ones for the algorithm – think of quick-sort for example.

    (b) If I prove that an algorithm takes $\Theta(n^2)$ worst-case time, is it possible that it takes $O(n)$ on all inputs?
    No. This conflicts with it being $\Omega(n^2)$.

    (c) Is the function $f(n) = \Theta(n^2)$, where

    $$f(n) = \begin{cases} 100n^2 & \text{for even } n \\ 20n^2 - n & \text{for odd } n \end{cases}$$

    Yes since

    $$n^2 \leq f(n) \leq 200n^2 \text{ for all } n$$

3. **Problem 3-1:**
$p(n) = a_0 + a_1 n + ... + a_d n^d; a_d > 0$. Since $a_d > 0$, $\exists n_1$ such that $p(n) > 0 \forall n \geq n_1$. (This is required for all parts of this question.)

$$\lim_{n \to \infty} \frac{p(n)}{n^k} = \begin{cases} 0 \text{ if } k > d \\ a_d \text{ if } k = d \\ \infty \text{ if } k < d \end{cases}$$

Parts (d) and (e) follow from this directly.

To show that $p(n) = O(n^k)$ for $k \geq d$, take $c_2 = 2a_d$. Since the above limit is no more than a_d, we have the desired result.

To show that $p(n) = \Omega(n^k)$ for $k \leq d$, take $c_1 = \frac{1}{2}a_d$. Since the above limit is no less than a_d, we have the desired result.

The part for $k = d$, follows from the above two results.

4. **Problem 3.2:** $k \geq 1; \epsilon > 0; c > 1$

	$f(n) = A$	$g(n) = B$	O	o	Ω	ω	Θ
a	$\lg^k n$	n^ϵ	YES	YES	NO	NO	NO
b	n^k	c^n	YES	YES	NO	NO	NO
c	$\sqrt{n}$	$n^{\sin n}$	NO	NO	NO	NO	NO
d	2^n	$2^{\frac{n}{2}}$	NO	NO	YES	YES	NO
e	$n^{\lg m}$	$m^{\ln n}$	YES	NO	YES	NO	YES
f	$\lg(n!)$	$\lg(n^n)$	YES	NO	YES	NO	YES

Parts (a),(b), and (f) were done in class. Part (d) is the same as #1(ii). We now discuss the details of parts (c) and (e) below:

c $f(n) = n^{\frac{1}{2}}$; $g(n)$ is a power of n whose value oscillates between -1 and +1. Hence nothing can be said about this pair with respect to order notation.

e $n^{\lg m} = (2^{\lg n})^{\lg m} = (2^{\lg m})^{\lg n} = m^{\lg n}$. Hence in this case Θ applies and hence both O and Ω also apply.