Due at the start of class Wednesday, June 11, 2003.

**Problem 1.** Professor Hilbert invents a program that uses exactly $1000n^3$ operations to solve the borogove problem of size $n$. Professor Poincare invents a program that uses $1.5^n/10^5$ operations to solve the same problem.

(a) Assume that you are willing to run Hilbert’s program for some specific amount of time. Assume that your computer speed increases by a factor of 10. How much larger a problem can you solve in the same amount of time? Show your work.

(b) Assume that your computer executes one billion operations per second. Approximately how large a problem can you solve in a year? Show your work (but not the arithmetic).

(c) Assume that you are willing to run Poincare’s program for some specific amount of time. Assume that your computer speed increases by a factor of 10. How much larger a problem can you solve in the same amount of time? Show your work.

(d) Assume that your computer still executes one billion operations per second. Approximately how large a problem can you now solve in a year? Show your work (but not the arithmetic).

(e) Whose program should you use if you are going to run it for a year?

(f) For approximately what values of $n$ is Professor Hilbert’s program faster than Professor Poincare’s? Show your work.

**Problem 2.** Show the following. In each of (a), (b), and (c) state specific values of the constants (e.g. $c_1$, $c_2$, $n_0$) you used to satisfy the conditions, and show how you arrived at the values.

(a) $2n^2 + 2n + 6 = \Theta(n^2)$
(b) $4n^3 - 3n^2 + 8 = \Theta(n^3)$
(c) $2n^2 + 5n(\log n)^3 = O(n^2)$ [Hint: Find $n_0$ such that $(\log n)^3 \leq n$, for all $n \geq n_0$.]
(d) $n \log n = \theta(n^{1.5})$

**Problem 3.** For each pair of expressions $(A, B)$ below, indicate whether $A$ is $O$, $o$, $\Omega$, $\omega$, or $\Theta$ of $B$. Note that zero, one or more of these relations may hold for a given pair; list all correct ones.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n^{100}$</td>
<td>$2^n$</td>
</tr>
<tr>
<td>$\log n + 12$</td>
<td>$\sqrt{n}$</td>
</tr>
<tr>
<td>$\sqrt{n}$</td>
<td>$n^{\cos(\pi n/8)}$</td>
</tr>
<tr>
<td>$10^n$</td>
<td>$100^n$</td>
</tr>
<tr>
<td>$n \log n$</td>
<td>$(\log n)^n$</td>
</tr>
<tr>
<td>$\log(n!)$</td>
<td>$n \log n$</td>
</tr>
</tbody>
</table>
Problem 4. Consider the insertion sort algorithm. (Either version will suffice.)

(a) Give a double summation for the number of moves in the worst case.
(b) Simplify the summation.
(c) Give a double summation for the number of moves in the average case.
(d) Simplify the summation.