

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.

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

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

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

- | | A | B |
|-----|-----------------|---------------------|
| (a) | n^{100} | 2^n |
| (b) | $(\log n)^{12}$ | \sqrt{n} |
| (c) | \sqrt{n} | $n^{\cos(\pi n/8)}$ |
| (d) | 10^n | 100^n |
| (e) | $n^{\log n}$ | $(\log n)^n$ |
| (f) | $\log(n!)$ | $n \log n$ |

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.