Due at the start of class Thursday, Feb 26, 2004.

Problem 1. In each case 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) $2n^3 - 9n^2 + 8 = \Theta(n^3)$

(b) $7n^2 - 10n = \Theta(n^2)$.

Problem 2. Design an algorithm to find the smallest and second smallest elements using at most $n - 1 + \log_2 n$ comparisons. If you like you may assume that n is a power of 2.

Problem 3. The *mode* of a set of numbers is the number that occurs most frequently in the set. The set $\{4, 3, 6, 4, 3, 4, 1\}$ has mode 4. Give an efficient algorithm to compute the mode of a set. Also give the running time of your algorithm.

Suppose we know that there exists an element that occurs at least $\frac{n}{2} + 1$ times in the set. Give an $O(n)$ algorithm to find the mode.

Problem 4. Prove that $f(n) = 3n^2 + 4n + 17 \in O(n^3)$. Also prove that $f(n) \in O(n^2)$.

Problem 5. You are given n nuts and n matching bolts in a jumbled heap. In each operation you are allowed to take one nut and one bolt and to check if they fit. You can conclude that either you have found the matching nut for the bolt, or the nut is too big or too small. (You cannot compare two nuts or two bolts directly.) Design a "good" algorithm to find the matching pairs of nuts and bolts. How many operations does your algorithm perform?