Due at the start of class Wednesday, November 9, 2011.

**Problem 1.** Illustrate the operation of radix sort on the following list of English words:

ilk, elf, jig, leg, elk, fig, ill, gel, eke, egg, gig, ell

**Problem 2.** Suppose that you are given \( n \) red and \( n \) blue water jugs, all of different shapes and sizes. All red jugs hold different amounts of water, as do all blue ones. Moreover, for every red jug, there is a blue jug that holds the same amount of water, and vice versa.

It is your task to find a grouping of the jugs into pairs of red and blue jugs that hold the same amount of water. To do so, you may perform the following operation: pick a pair of jugs in which one is red and one is blue, fill the red jug with water, and then pour the water into the blue jug. This operation will tell you whether the red or the blue jug can hold more water, or if they are of the same volume. Assume that such a comparison takes one time unit. Your goal is to find an algorithm that makes a minimum number of comparisons to determine the grouping. Remember that you may not directly compare two red jugs or two blue jugs.

(a) Describe a deterministic algorithm that uses \( \Theta(n^2) \) comparisons to group the jugs into pairs.

(b) Prove a lower bound of \( \Omega(n \log n) \) for the number of comparisons an algorithm solving this problem must make.

(c) Give a randomized algorithm whose expected number of comparisons is \( O(n \log n) \). Justify its running time.

(d) What is the worst-case number of comparisons for your algorithm. Justify.

**Problem 3.** Assume you have an algorithm that finds the median of \( n \) elements in \( cn \lg \lg n \) comparison steps (for some constant \( c \)).

(a) Give an efficient algorithm for selection based on this. It will, of course, not be linear time.

(b) How many comparisons does it use. Get the high order term exactly.

(c) Why might it be a good algorithm despite not being linear time?

**Problem 4.** Make an intuitive argument for why no algorithm for selection can use fewer than (about) \( 2n \) comparisons in the worst case. [If you feel that \( 2n \) is not a good choice, but think that you can make an intelligent argument for some other reasonable value, such as \( 1.5n \) or \( 2.5n \), then do so. (An argument for just \( n - 1 \) is obvious.)]