Due at the start of class, Wednesday, June 30.

Problem 1. Assume you have a list of n elements where the first n/k elements are the smallest (but not sorted), the next group of n/k elements are the next smallest (but not sorted), ..., and the last n/k elements are the largest (but not sorted). You may assume k divides n.

(a) Give an algorithm that sorts this list with as few comparisons as possible (as a function of n and k). Just get the high order term right. How many comparisons does your algorithm use?

(b) Show that your algorithm is optimal using a decision tree argument on the *entire* list. (I.e., do not argue that you must solve k independent sorting problems.)

Problem 2.

(a) Illustrate the operation of radix sort on the following list of English words: RUTS, TOPS, SUNS, SPOT, TONS, OPTS, TORS, SOTS, ROOT, OUTS, SUPS, PUTT

(b) Write an English sentence using both "tor" and "sot" (that indicates you understand the meanings of both words).

Problem 3. In class, we solved the selection problem by breaking the list into groups of 5 elements each.

(a) We used the fact that you can find the median of 5 numbers with 10 comparisons (by sorting). It turns out that you can find the median with only 6 comparisons.

  (i) Write down the recurrence for the running time using this new fact. (You can ignore floors and ceilings, as we did in class.)

  (ii) Solve the recurrence.

(b) (i) How may comparisons do you need to find the median of 3 elements? Justify your answer.

  (ii) Write down the recurrence for a selection algorithm based on columns with three elements each. (You can ignore floors and ceilings, as we did in class.)

  (iii) Solve the recurrence.

(c) You need to 10 comparisons to find the median of 7 elements?

  (i) Write down the recurrence for a selection algorithm based on columns with 7 elements each. (You can ignore floors and ceilings, as we did in class.)

  (ii) Solve the recurrence.

(d) What did you learn?

Problem 4. Assume we have n distinct elements $x_1, x_2, \ldots, x_n$ with non-negative weights $w_1, w_2, \ldots, w_n$, such that $\sum_{i=1}^{n} w_i = 1$. The *weighted middle element* is the element x_k satisfying

$$\sum_{x_i < x_k} w_i < \frac{1}{2} \quad \text{and} \quad \sum_{x_i > x_k} w_i \leq \frac{1}{2}$$

(a) Show how to compute the weighted middle element in $\Theta(n \log n)$ time using sorting.

(b) Show how to compute the weighted middle element in $\Theta(n)$ time using a linear time selection algorithm.

Problem 5. Challenge problem. Show how to find the median of 5 numbers with only 6 comparisons.