Due at the start of class Wednesday, June 15, 2011.

Problem 1. Consider the following recurrence, defined for n a power of 5:

$$T(n) = \begin{cases} 19 & \text{if } n = 1 \\ 3T(n/5) + n - 4 & \text{otherwise} \end{cases}$$

(a) Solve the recurrence exactly using the iteration method. Simplify as much as possible.

(b) Use mathematical induction to verify your solution.

Problem 2. Use the formulas derived in class to obtain exact solutions to the following two recurrences.

(a) Let n be a power of 2.

$$T(n) = \begin{cases} 4 & \text{if } n = 1 \\ 5T(n/2) + 3n^2 & \text{otherwise} \end{cases}$$

(b) Let n be a power of 4.

$$T(n) = \begin{cases} 3 & \text{if } n = 1 \\ 2T(n/4) + 4n + 1 & \text{otherwise} \end{cases}$$

Problem 3. Consider the following recurrence.

$$T(n) = \begin{cases} 0 & \text{if } n = 0 \\ T(\lfloor n/2 \rfloor) + T(\lfloor n/4 \rfloor) + 3n & \text{otherwise} \end{cases}$$

Use constructive induction to find a constant c such that $T(n) \leq cn$.

Problem 4. Selection Sort can be thought of as a recursive algorithm as follows: Find the largest element and put it at the end of the list (to be sorted). Recursively sort the remaining elements.

(a) Write down the recursive version of Selection Sort in psuedocode.

(b) Derive a recurrence for the exact number of comparisons the algorithm uses.

(c) Use the iteration method to solve the recurrence. Simplify as much as possible.