Due by 2:30pm, Friday, September 28.

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

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

Solve the recurrence exactly using the iteration method (as done in class). Simplify as much as possible.

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 idea for a sorting algorithm, similar to Selection Sort: Iterate through the array keeping track of the two largest elements. Put these two elements at the end of the array. Then look for the next two largest elements, putting them at the end of the "remaining" array. Etc. You may assume that n is even and $n \geq 2$.

(a) Give the psuedocode for an iterative version of this algorithm.

(b) Analyze its best case number of comparisons. Try to do it with summations. State what the best case is.

(c) Analyze its worst case number of comparisons. Try to do it with summations. State what the worst case is.

(d) Give the psuedocode for a recursive version of this algorithm.

(e) Derive a recurrence for the exact number of comparisons the algorithm uses in the worst case.

(f) Solve the recurrence exactly using the iteration method (as done in class). Simplify as much as possible. You should get the same answer as in Part (c).

"Master Theorem":

$$T(n) = \begin{cases} aT(n/b) + cn^d & n > 1 \\ f & n = 1 \end{cases}$$

implies

$$T(n) = \begin{cases} \left(f + \frac{c}{ab^{-d}-1}\right) n^{\log_b a} - \frac{cn^d}{ab^{-d}-1} = \begin{cases} \Theta(n^{\log_b a}) & a > b^d \\ \Theta(n^d) & a < b^d \end{cases} \\ n^d(f + c\log_b n) = \Theta(n^d \log_b n) & a = b^d \end{cases}.$$

`Summing solutions:` If

$$T(n) = \begin{cases} aT(n/b) + \sum c_i n^{d_i} & n > 1 \\ f & n = 1 \end{cases}$$

then we can just sum the solutions of each recurrence:

$$T_i(n) = \begin{cases} aT_i(n/b) + c_i n^{d_i} & n > 1 \\ 0 & n = 1 \end{cases}$$

and add in $fn^{\log_b a}$ for the contribution from the leaves.