CSCI-GA.1170-001/002 Fundamental Algorithms	September 10, 2014
Problem Set 2	
Lecturer: Yevgeniy Dodis	*Due: Wednesday, September 17*

Problem 2-1 (Mergesort) 10 points

(a) (6 points) Suppose you have some procedure FASTMERGE that given two sorted lists of length m each, merges them into one sorted list using m^c steps for some constant $c > 0$. Write a recursive algorithm using FASTMERGE to sort a list of length n and also calculate the run-time of this algorithm as a function of c. For what values of c does the algorithm perform better than $O(n \log n)$.

(b) (4 points) Let $A[1 \dots n]$ be an array such that the first $n - \sqrt{n}$ elements are already in sorted order. Write an algorithm that will sort A in substantially better than $O(n \log n)$ steps.

Problem 2-2 (Functionality vs. Running Time) 10 points

Consider the following recursive procedure.

Bla(n):
  If $n = 1$ **Then Return** 1
  Else Return Bla($n/2$) + Bla($n/2$) + Bla($n/2$)

(a) (3 points) What function of n does Bla compute (assume it is always called on n which is a power of 2)?

(b) (3 points) What is the running time $T(n)$ of Bla?

(c) (4 points) How do the answers to (a) and (b) change if we replace the last line by "**Else Return** $3 \cdot$ Bla($n/2$)"?

Problem 2-3 (Different Methods for Recurrences) 14 points

Consider the recurrence $T(n) = 8T(n/4) + n$ with initial condition $T(1) = 1$.

(a) (2 points) Solve it asymptotically using the "master theorem".

(b) (4 points) Solve it by the "guess-then-verify method". Namely, guess a function $g(n)$ — presumably solving part (a) will give you a good guess — and argue by induction that for all values of n we have $T(n) \leq g(n)$. What is the "smallest" $g(n)$ for which your inductive proof works?

(c) (4 points) Solve it by the "recursive tree method". Namely, draw the full recursive tree for this recurrence, and sum up all the value to get the final time estimate. Again, try to be as precise as you can (i.e., asymptotic answer is OK, but would be nice if you preserve a "leading constant" as well).

(d) (4 points) Solve it *precisely* using the "domain-range substitution" technique. Namely, make several changes of variables until you get a basic recurrence of the form $S(k) = S(k-1) + f(k)$ for some f, and then compute the answer from there. Make sure you carefully maintain the correct initial condition.

(e) This part will not be graded. However, briefly describe your personal comparison of the above 4 methods. Which one was the fastest? The easiest? The most precise?

Problem 2-4 (More Recurrences) 12 points

Solve the following recurrences using any method you like. If you use "master theorem", use the version from the book and justify why it applies. Assume $T(1) = 2$, and be sure you explain every important step.

(a) (3 points) $T(n) = T(2n/3) + \sqrt{n}$.

(b) (4 points) $T(n) = T(n/2) + \log n$.

(c) (5 points) $T(n) = T(\sqrt{n}) + 1$. (**Hint**: Substitute ... until you are done!)