Problem Set 1—Due Friday, April 7

Instructions: Write up your your solutions as clearly and succinctly as you can. Don't forget to acknowledge anyone with whom you discussed problems.

Problem 1. Recall Protocol "B" for cake cutting (from the first lecture of class): when there are $n = 2$ players we use the standard solution, and when there are $n \geq 3$ players —

Player 1 cuts the cake into n equal pieces;
Players $2, \ldots, n$ each (in turn) choose the largest available piece;
Players 1 takes the remaining piece and goes home;
Players $2, \ldots n$ recombine their pieces and (recursively) solve the cake cutting problem amongst themselves.

Prove or disprove: *Algorithm B is a fair cake cutting protocol.* (Recall that a cake cutting protocol is *fair* if each player is guaranteed at least $1/n$ of the cake, assuming he plays honestly).

Problem 2. Suppose you had a choice of four algorithms to solve a problem, with the following approximate running time as a function of input size n:

Algorithm 1 takes	2^n seconds
Algorithm 2 takes	$20n^2$ seconds
Algorithm 3 takes	$1000n \lg n$ seconds
Algorithm 4 takes	$10000n$ seconds

Give the range of n for which each algorithm is optimal.

Problem 3. Let A be an array of n integers such that $A[1] < A[2] < \cdots < A[n]$. Specify an $O(\lg n)$ divide-and-conquer algorithm which either finds an $i \in \{1, \ldots, n\}$ such that $A[i] = i$ or else determines that there is no such i. Justify the correctness and running time of your algorithm.

Problem 4. You are given an array $A = (A[1], \ldots, A[n])$ of n real numbers. You want to determine the maximum and minimum value in A. You do this by comparing elements of A. Assume $n = 2^d$ for some $d \geq 1$.

(a) The "obvious" algorithm makes $2n - 2$ comparisons. Explain.

(b) Carefully specify a more efficient divide-and-conquer algorithm.

(c) Let $T(n) =$ the number of comparisons your algorithm makes when the input array has n elements. Write a recurrence relation for $T(n)$. (Recall that $n \geq 2$ is a power of 2.)

(d) Show that your recurrence relation has as its solution $T(n) = 1.5n - 2$.
Hint: Use induction.
Comment: If your $T(n)$ is larger than this, you need to improve your algorithm. If your $T(n)$ is smaller than this, your algorithm is incorrect.)