Problem Set 2—Due Thursday, 8 July 1999

Problem 4. Solve the following recurrences, getting $\Theta(f(n))$ answers. Assume that $T(n)$ is constant for all sufficiently small n.

A. $T(n) = T(9n/10) + n$

B. $T(n) = 2T(n/4) + \sqrt{n}$

C. $T(n) = T(\sqrt{n}) + 1$

D. $T(n) = 2T(n/2) + n/\lg n$

E. $T(n) = T(n-1) + \lg n$.

Problem 5. Recalling the analysis of Mergesort, let $S(1) = S(0) = 0$ and let $S(n) = S(\lfloor n/2 \rfloor) + S(\lceil n/2 \rceil) + n - 1$ for $n \geq 2$. Show how, for any n, to construct a worst-case instance of Mergesort: the instance takes $S(n)$ comparisons. Show the sequence you construct for $n = 8$ and $n = 13$.

Problem 6. Aj. Darunee has n supposedly identical VLSI chips. Unfortunately, some of the chips are *bad*—they don't do what they are supposed to do. Fewer than $n/2$ of the chips are bad. Aj. Darunee wants to identify which chips are good and which are bad. To do this she has some testing machine. It takes two chips and they test each other. Each chip reports if the other chip is good or bad. If a chip is good it correctly reports about the other chip. If a chip is bad it might report anything about the other chip. Thus there are four possible outcomes for running the test: gg, gb, bg, bb, indicating what chip A says about chip B (good or bad) and what chip B says about chip A (good or bad). Clearly a result of gg indicates that both chips are good, while a result of gb, bg, or bb indicates that at least one of the two chips is bad.

Show that $\Theta(n)$ tests are sufficient to identify which chips are good and which chips are bad. *Hint: First try to find a single good chip. For this problem, show that $n/2$ tests are sufficient to reduce the problem to one of half the size.*

Problem 7. In lecture we gave an analysis of the determinisitic selection algorithm in which we treated the input as an array of columns of 5 objects, and we saw the resulting analysis yielded a bound of $16n$ comparisons. Formulate the analagous algorithm using columns of 3 objects instead of 5, and given a good bound on the number of comparisons for your algorithm. Use the fact that we can find the median of 3 elements in 3 comparisons. Now formulate the analagous algorithm using columns of 7 objects instead of 5, and given a good bound on the number of comparisons for your algorithm. Use the fact that we can find the median of 7 elements in 10 comparisons.

Problem 8. You are given a box containing nuts and bolts. Each nut fits one and only one bolt. You want to sort the nuts and bolts. The only technique that you may use to assess the relative sizes of the nuts (and bolts) is to try to screw a particular nut N_i onto a particular bolt B_j. If the nut is too big for the bolt it will go on loosely, and you can tell that. If the nut is too small for the bolt it it won't go on at all, and you can tell that. And if the nut exactly fits the bolt, you can tell that, too.

Give an algorithm to solve the Nuts-and-Bolts problem in $O(n \lg n)$ expected time. Why does your algorithm run in the stated time bound? *Hint: think of Quicksort.*