Problem Set 2—Due Friday, April 14

Problem 5.

(a) Rank the following functions by order of growth. That is, find an arrangement of the functions $g_1, \ldots, g_{25}$ such that $g_1 = \Omega(g_2)$, $g_2 = \Omega(g_3)$, $\ldots$, $g_{24} = \Omega(g_{25})$. Partition your list into equivalence classes such that $f(n)$ and $g(n)$ are in the same class if and only if $f(n) = \Theta(g(n))$. It is not necessary to show your work.

$\lg n$	$\ln n$	$\lg\lg n$	$\ln^2 n$	$\ln\ln n$
$\lg^2 n$	$\lg(n!)$	$2^{\sqrt{\lg n}}$	$n!$	$n^{0.001}$
1	n	e^n	$2n^2$	2^n
$2^{\sqrt{n}}$	$n\lg n$	$4^{\lg n}$	$n^{\lg 7}$	$n^{\log n}$
$n^{\lg\lg n}$	2^{2^n}	n^2	n^3	$(2/3)^n$

(b) Give an example of a function $f(n)$ such that for all $g_i(n)$ above, $f(n)$ is neither $O(g_i(n))$ nor $\Omega(g_i(n))$.

(c) Can you do part (b) with a monotonic function f? (Explain.) *For experts only*: Drop the slowest growing pair of functions from the list. Now can you do part (b) with a monotonic function? (Explain.)

Problem 6. Do the "even letters" (b,d,f,h) of Problem 4–1 (p. 72), and the "even letters" (b,d,f) of Problem 4–4 (pp. 73–74).

Problem 7. In class we defined $S(n)$ to be the maximum, over all sequences of length n, of the number of comparisons used to sort n distinct items using MERGESORT. We then said that $S(1) = 0$ and $S(n) = S(\lfloor n/2 \rfloor) + S(\lceil n/2 \rceil) + n - 1$ for $n \geq 2$. This has not been justified; all that is clear is that the defined quantity is $\leq$ the given recurrence. Demonstrate equality. That is, for each $n \geq 1$ construct a sequence of n numbers (make them $1, 2, \ldots, n$) such that MERGESORT really *does* use the number of comparisons given by the recurrence relation. Give a concrete example to explain how your contraction works: show the sequence you get for $n = 8$.

Problem 8. Find a version of MERGESORT, say MERGESORT*, which uses fewer comparisons (in the worst case) for infinitely many n (and the same number of comparisons otherwise). Your algorithm should continue to look like MERGESORT: for small n, solve the problem directly; else break the problem into subproblems of size $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil$, recurse, and merge. Write an exact recurrence relation for the worst case number of comparisons that your MERGESORT* uses. Tabulate the worst case number of comparisons that MERGESORT and MERGESORT* use for $n = 1, 2, \ldots, 15$.

Problem 9. Problem 4-7, page 75.