Cosi 30b Quiz #1 21 February, 1997

Suggested times are 1:15, 2:5, 3:15, 4:15, 5:15, 6:10. Start with the questions you know you can answer.

1. For each of the following algorithms, give a recurrence for the worst-case running time, state its solution, and prove it by induction: (a) Binary Search, (b) Merge Sort.

   (a) Basis: $T(n) = c' + T(n/2)$; $T(n) = \Theta(\lg(n))$.
   $T(1) = \Theta(1) = \Theta(\log(1))$; Assume $T(m) \leq c\lg(m)$ for all $m < n$. Then $T(n) \leq c' + c\lg(n/2) = c' + c\lg(n) - c$, and we can choose c large enough to make $T(n) \leq c\lg(n)$. This works out more "neatly" if you simply set all the c's $= 1$.

   (b) Basis: $T(1) = c', T(n) = c'n + 2T(n/2)$; $T(n) = \Theta(n\lg(n))$
   $T(1) = c'$; assume $T(m) \leq cm\lg(m)$ for $m < n$. Then $T(n) \leq c'n + 2c(n/2)\lg(n/2) = c''n + cn\lg(n) = \Theta(n)$

2. (a) List the *inversions* in the array {2,1,3,5,4}: (2,1) and (5,4).

   (b) What arrangement of the integers [1..n] has the most inversions? The reversal sort. How many? $\binom{n}{2} = n(n-1)/2$.

3. (a) What is the asymptotic running time of Heap Sort on an input array that is already sorted in decreasing order? $n\lg(n)$ In increasing order? $n\lg(n)$

   (b) In a heap containing the integers $[1..2^k - 1]$, what is the smallest number that can appear as the root of the left subtree? $2^{k-1} - 1$.

   (c) If the keys are the distinct integers [1..n], for what pivot values will CL&R's *Partition(A,lo,hi)* yield one subarray of size 1 and the other of size n-1 ? 1, 2, and $n - 1$.

   (d) How many swaps will CL&R's Quick Sort make when all 2^k elements of an input array have the same value? $(n/2)\lg(n)$

4. (a) What is the smallest depth (distance from the root) of any leaf in a *decision tree* for comparison sorting? $n - 1$

   (b) Is is possible for a comparison sort to run in linear time for a fraction of inputs at least the square root of all inputs? No. What about for a fraction of inputs at least the log of all inputs? Yes. You have to see how 2^{cn} compares with $\sqrt{(n!)}$ in the first case, and $\lg(n!)$ in the second. To do the second, take logs of both sides and see the asymptotic order of growth.

   (c) Give the smallest example for which Radix Sort fails if done from most-significant to least-significant digit. [12, 21].

5. In Bucket Sort assume that each of n keys $k_1, k_2, \ldots, k_n$ is independently, equally likely to fall into each of n buckets $b_1, b_2, \ldots, b_n$. Hint: use the case n=3 for guidance.

(a) In how many ways can no bucket be empty? $n!$ What is the chance of this happening? $n!/n^n$

(b) In how many ways can exactly one bucket be empty? This was the hardest question on the exam: There are n ways to choose the empty bucket; there are then $n-1$ ways to pick the bucket with two balls, and $\binom{n}{2}$ ways to pick the ball to go in it. Finally, there are $(n-2)!$ ways to distribute the remaining balls in the $n-2$ remaining buckets with none empty. (See part (a)). So the chance is $n(n-1)\binom{n}{2}(n-2)!/n^n$.

(c) In how many ways can all the keys fall into a single bucket? $n = \binom{n}{1}1^n(n-1)^0$ What is the chance of this happening? $n^{-(n-1)}$

6. Suppose the linear *Median* algorithm ("Five Guys Algorithm") breaks inputs into groups of seven, rather than of five. Give a recurrence for its worst-case running time, and prove that the time is $O(n)$. (Don't forget a basis for the proof.)
$T(n) = \Theta(n) + T(n/7) + T(5n/7)$ The proof is just like the one in CL&R.