Math 55 — Discrete Mathematics — Spring 2003

Quiz 10 Solutions

OK, I admit this turned out to be too hard for a quiz problem. We'll put an asterisk by this quiz score and probably count it less than the others.

However, please study the solution—I think you will be able to understand it, even if you didn't know how to solve it yourself, and it contains some important ideas about how to analyze recursive algorithms.

Version 1:

Suppose we are given an algorithm $mid(a_1, \ldots, a_n)$ that returns an element in the middle third of a list of distinct numbers $a_1, \ldots, a_n$. Precisely, it returns an index i such that at least $\lfloor n/3 \rfloor$ of the elements a_j are less than a_i and at least $\lfloor n/3 \rfloor$ of them are greater than a_i.

Consider the following variant of the quicksort algorithm:

procedure $qsort(a_1, \ldots, a_n)$
if $n = 0$ **return** $\emptyset$ (the empty list)
if $n = 1$ **return** a_1 (a list of one element)
else
    $i \coloneqq mid(a_1, \ldots, a_n)$
    $b_1, \ldots, b_k \coloneqq$ list of elements $a_j < a_i$
    $c_1, \ldots, c_l \coloneqq$ list of elements $a_j > a_i$
    return $qsort(b_1, \ldots, b_k)$, a_i, $qsort(c_1, \ldots, c_l)$

(a) Show that the algorithm is correct, *i.e.*, the output is the input list sorted into increasing order. Assume that the elements of the input list are distinct.

(b) Suppose there is a constant C such that procedure *mid* uses no more than Cn comparisons for an input list of length n. Prove that procedure *qsort* uses no more than $(C+1)n\log_{3/2}(n+1)$ comparisons to sort an input list of length n, for all $n \geq 0$.

Hint: show that $k+1$ and $l+1$ are less than or equal to $2(n+1)/3$.

Version 2 was the same with $\lfloor n/4 \rfloor$ in place of $\lfloor n/3 \rfloor$ and $\log_{4/3}$ in place of $\log_{3/2}$. I'll give the solution for Version 1. For Version 2 it's the same with some 2's and 3's changed to 3's and 4's.

(a) The algorithm is clearly correct for $n = 0$ and $n = 1$. For $n > 1$ we can assume by induction that it is correct for inputs of length less than n. Since k and l are less than n, the output is the sorted list b, followed by a_i, followed by the sorted list c. This is correct since b contains all the numbers less than a_i and c contains all those greater than a_i.

(b) For $n = 0$ and $n = 1$, procedure *qsort* uses zero comparisons, and $0 \leq (C+1)n\log_{3/2}(n+1)$. (We can assume $C \geq 0$, of course.)

For $n > 1$, assume by induction that the upper bound holds for smaller input lists. The recursive calls take at most

$$(C+1)k\log_{3/2}(k+1) + (C+1)l\log_{3/2}(l+1)$$

comparisons. We need an additional $k + l$ comparisons to separate the lists b and c, and at most Cn comparisons for the call to *mid*, for a total of at most

$$(C+1)k\log_{3/2}(k+1) + (C+1)l\log_{3/2}(l+1) + k + l + Cn. \tag{1}$$

Following the hint, we have $k, l \geq \lfloor n/3 \rfloor$, and $k + l = n - 1$, so $k \leq n - 1 - \lfloor n/3 \rfloor$, and $k + 1 \leq n - \lfloor n/3 \rfloor \leq (2n+2)/3 = 2(n+1)/3$. Similarly, $l + 1 \leq 2(n+1)/3$. Hence

$$\log_{3/2}(k+1) \leq \log_{3/2} \frac{2}{3}(n+1) = \log_{3/2}(n+1) - 1,$$

and similarly

$$\log_{3/2}(l+1) \leq \log_{3/2}(n+1) - 1.$$

Formula (1) above is therefore less than or equal to

$$(C+1)(k+l)(\log_{3/2}(n+1) - 1) + k + l + Cn.$$

Using the fact that $k + l < n$, this is less than

$$(C+1)n(\log_{3/2}(n+1) - 1) + n + Cn = (C+1)n\log_{3/2}(n+1).$$