CSCI 3110 mini-Assignment 7 Solutions

December 5, 2012

Ex.1 2.18 Consider the task of searching a sorted array $A[1...n]$ for a given element x: a task we usually perform by binary search in time $O(\log n)$. Show that any algorithm that accesses the array only via comparisons (that is, by asking questions of the form is $A[i] \leq z$?), must take $\Omega(\log n)$ steps.

Consider the decision tree for the problem. At each internal node a comparision occurs $A[i] \leq x$, the result of which a decision is made for the next part of the search . The leaves of the decision tree represent the possible outputs of the algorithm: Here they are the n indices of the elements plus the possibility that the element x is not present in A. This means the decision tree must have at least $n+1$ leaves. A path from root to a leaf represents an execution of the algorithm. Since the tree has at least $(n+1)$ leaves, the decision tree must have height at least $\log(n+1)$, giving an execution time in $\Omega(\log(n+1))$.

Ex. 2. In the algorithm SELECT covered in class, the input is divided into groups of 5. Will the algorithm work in linear time if they are divided into (i) groups of 7? (ii) Give an argument that SELECT will not run in linear time if groups of 3 are used. (Bonus: if you can obtain a general expression, then you could answer both questions).

SOLUTION 1:
(i) Yes, the algorithm will work in linear time if they are divided into groups of 7. There are $n/7$ groups with at least $4 \cdot 1/2 \cdot (n/7)$ elements that are less than or equal to the median of the medians, and at least as many that are greater than or equal to the median of medians. Thus, the larger subset after partitioning has at most $n - 2n/7 = 5n/7$ elements. The running time is $T(n) = T(n/7) + T(5n/7) + \Theta(n)$ Solve using substitution:
Assume:

$$T(k) \leq c^* k \; for \; k < n$$

$$T(k) \leq c^*(n/7) + c^*(5n/7) + c_1 n = c^*(6n/7) + c_1 n = c^* n + (c_1 - c^*(1/7))n \leq c^* n$$

which is linear if $(c_1 - c^*(1/7)) \leq 0$ (for a more careful analysis, see class notes on SELECT (attached).

(ii) Using a similar analysis we get that there are only $2 \cdot 1/2 \cdot (n/3)$ elements that are guaranteed to be larger or smaller than the median of medians. The recurrence is
$T(n) = T(n/3) + T(2n/3) + \Theta(n) = \Theta(n \log n)$. (solve using substitution)
SOLUTION 2:
If we use k elements as a group, the number of elements less than the median is:
$\lceil k/2 \rceil \left(\lceil \frac{1}{2} \lceil \frac{n}{k} \rceil \rceil - 2 \right) \geq \frac{n}{4} - k$. In the worst case, we need to recursively call Select for $n - \left(\frac{n}{4} - k \right) = \frac{3n}{4} + k$ times. Thus we have:

$$T(n) = T\left(\lceil \frac{n}{k} \rceil \right) + T\left(\frac{3n}{4} + k \right) + O(n)$$

Using iterations, we have:

$$
\begin{aligned}
T(n) &\leq c\left(\lceil \frac{n}{k} \rceil\right) + c\left(\frac{3n}{4} + k\right) + O(n) \\
&\leq c\left(\frac{n}{k} + 1\right) + \frac{3cn}{4} + ck + O(n) \\
&= \frac{cn}{k} + \frac{3cn}{4} + c(k+1) + O(n) \\
&= cn(\frac{1}{k} + \frac{3}{4}) + c(k+1) + O(n) \\
&\leq cn
\end{aligned}
$$

where $\frac{1}{k} + \frac{3}{4} < 1 => k > 4$

Hence, the group size must be larger than 4 to make it linear.