Technische Universität München
Institut für Informatik
Lehrstuhl für Effiziente Algorithmen
Lehrstuhl für numerische Programmierung

WS 2002/2003
Worksheet 6
2.12.2002

Fundamental Algorithms

Exercise 1

– obsolete –

Exercise 2

Consider a partitioning algorithm that, in the worst case, will partition an array of m elements into two partitions of size $\lfloor \epsilon m \rfloor$ and $\lceil (1-\epsilon)m \rceil$, where ϵ is fixed, and $0 < \epsilon < 1$. Show that a quicksort algorithm based on this partitioning has a worst-case complexity of $O(n \log n)$.

Solution:

Again, we will only count comparisons between array elements.

Using that the partitioning step will require at most n comparisons, we get the following recurrence for the necessary number $C(n)$ of comparisons:

$$\begin{aligned} C(1) &= 0 \\ C(n) &= C(\epsilon n) + C((1-\epsilon)n) + n \end{aligned}$$

We guess $C(n) := an \log_2 n + b$ as the solution, and try to find constants a and b such that the recurrence is satisfied:
case $n = 1$:

$$C(1) = a \cdot 1 \cdot \log_2 1 + b = 0 \quad \Leftrightarrow b = 0,$$

hence, $C(n) = an \log_2 n$.

case $n > 1$**:** We insert our guess into the recurrence:

$$
\begin{array}{rcl}
a n \log_2 n = C(n) & = & C(\epsilon n) + C((1-\epsilon)n) + n \\
\Leftrightarrow \quad a n \log_2 n & = & a\epsilon n \log_2(\epsilon n) + a(1-\epsilon)n \log_2((1-\epsilon)n) + n \\
\Leftrightarrow \quad a n \log_2 n & = & a\epsilon n \left(\log_2 \epsilon + \log_2 n\right) + a(1-\epsilon)n \left(\log_2(1-\epsilon) + \log_2 n\right) + n \\
\Leftrightarrow \quad a n \log_2 n & = & a\epsilon n \log_2 \epsilon + a\epsilon n \log_2 n + \\
 & & a(1-\epsilon)n \log_2(1-\epsilon) + a(1-\epsilon)n \log_2 n + n \\
\Leftrightarrow \quad a n \log_2 n & = & a\epsilon n \log_2 \epsilon + a\epsilon n \log_2 n + \\
 & & a n \log_2(1-\epsilon) - a\epsilon n \log_2(1-\epsilon) + a n \log_2 n - a\epsilon n \log_2 n + n \\
\Leftrightarrow \quad 0 & = & a\epsilon n \log_2 \epsilon + a n \log_2(1-\epsilon) - a\epsilon n \log_2(1-\epsilon) + n \\
\Leftrightarrow \quad 0 & = & a n \left(\epsilon \log_2 \epsilon + (1-\epsilon)\log_2(1-\epsilon)\right) + n \\
\Leftrightarrow \quad a & = & \dfrac{-1}{\epsilon \log_2 \epsilon + (1-\epsilon)\log_2(1-\epsilon)}
\end{array}
$$

Thus, the recurrence is satisfied if

$$C(n) = \frac{-n \log_2 n}{\epsilon \log_2 \epsilon + (1-\epsilon)\log_2(1-\epsilon)}$$

Note that the constant a will be very large for values of ϵ that are close to either 0 or 1. Thus, even very bad partitions will not destroy the $O(n \log n)$ complexity, provided that the respective partition sizes are bounded by ϵn and $(1-\epsilon)n$. However, bad partitions will still lead to slow algorithms due to the large constant factor involved.