Institut für Informatik
der Universität München
Dr. J. Johannsen
Dr. R. Matthes

SS 2003

29. April 2003

Übungen zur Vorlesung
Effiziente Algorithmen

Blatt 3

Aufgabe 9: (Einfachere Hälfte:) Geben Sie einen Algorithmus an, der mit $n + O(\log n)$ Vergleichen simultan das größte und zweitgrößte Element einer Liste von $n \geq 2$ verschiedenen Elementen findet.

(Schwierigere Hälfte:) Zeigen Sie, dass diese obere Schranke optimal ist: jedes Verfahren, das dies leistet, braucht im schlechtesten Fall mindestens $n + \lceil \log_2 n \rceil - 2$ Vergleiche.

Hinweis: Der gesuchte Algorithmus wird sogar höchstens $n+\lceil \log_2 n \rceil - 2$ Vergleiche brauchen. Um eine algorithmische Idee zu bekommen, kann man zunächst diese exakte *worst-case*-Vergleichszahl für $n \in \{2, 3, 4, 5, 6, 7, 8\}$ anstreben. Bei $n = 7$ oder spätestens $n = 8$ sollte man dann das allgemeine Prinzip erkennen und die obere Schranke beweisen können.

Aufgabe 10: Es sei ein Algorithmus gegeben, der das k-größte Element einer Liste von n Elementen nur durch paarweise Vergleiche bestimmt.

Zeigen Sie, dass derselbe Algorithmus ohne zusätzliche Vergleiche auch die $k-1$ größeren und die $n-k$ kleineren Elemente ausgeben könnte.

Aufgabe 11: Beim in der Vorlesung vorgestellten Algorithmus SELECT werden die Elemente der Eingabe in Gruppen von je 5 aufgeteilt.

Würde der Algorithmus auch in Linearzeit laufen, wenn man stattdessen in Gruppen von (a) je 3 oder (b) je 7 aufteilen würde? Falls ja, ist das jeweilige Verfahren besser oder schlechter als bei Aufteilung in Fünfergruppen?

Aufgabe 12: Verwenden Sie die Idee des Algorithmus SELECT, um eine Modifikation von QUICKSORT anzugeben, die im *worst-case* eine Laufzeit von $O(n \log n)$ erreicht.

Aufgabe 13: Angenommen, wir hätten schon eine fertige Routine, die in linearer Laufzeit nur den Median bestimmen kann. Benutzen Sie diese als "Black Box", um einen einfachen linearen Algorithmus für das (allgemeine) Selektionsproblem zu formulieren.