Institut für Informatik
der Universität München
Prof. M. Hofmann
Dr. J. Johannsen

SS 2002

3. Mai 2002

Übungen zur Vorlesung
Effiziente Algorithmen

Blatt 3

Aufgabe P-11: Geben Sie einen Algorithmus an, der simultan das größte und zweitgrößte Element einer Liste mit $n + O(\log n)$ Vergleichen findet.

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

Aufgabe P-12: 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 je 3 oder 7 aufteilen würde?
- Falls ja, ist das Verfahren besser oder schlechter als bei Aufteilung in Fünfergruppen?

Aufgabe P-13: Zeigen Sie, dass jedes Verfahren, das den Median einer Liste von n Zahlen nur durch paarweise Vergleiche bestimmt, mindestens $\lceil 3n/2 \rceil - O(\log n)$ Vergleiche durchführen muss.

Aufgabe P-14: Es sei ein Algorithmus gegeben, der das k-kleinste Element einer Liste von n Zahlen nur durch paarweise Vergleiche bestimmt.

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

Hausaufgaben:

Aufgabe H-9: 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. (4 Punkte)

Aufgabe H-10: Für eine Liste von n Zahlen sollen die k größten Elemente nach der Größe sortiert ausgegeben werden. Geben Sie für die beiden folgenden Methoden jeweils einen asymptotisch optimalen Algorithmus an, und analysieren Sie deren Laufzeit in Abhängigkeit von n und k:

- Erzeuge aus der Liste eine Prioritätsschlange, und wende k mal EXTRACT-MAX an.
- Verwende einen Algorithmus zum Auffinden des k-größten Elements, partitioniere die Liste mit diesem als Pivot, und sortiere die so gefundene Liste der k größten Elemente.

Vergleichen Sie die Ergebnisse. (8 Punkte)

Aufgabe H-11: Der Algorithmus SELECT aus der Vorlesung hat eine Laufzeit $T(n) = \Theta(n)$, jedoch mit einer sehr großen versteckten Konstante. Für kleine k kann man das k-größte Element besser bestimmenen:

- Beschreiben Sie einen Algorithmus zum Auffinden des k-größten Elements für $k \leq n/2$, der zunächst $\lfloor n/2 \rfloor$ disjunkte Paare vergleicht, und dann rekursiv die Menge der kleineren Elemente behandelt. Die Anzahl $U_k(n)$ der verwendeten Vergleiche sollte der folgenden Rekursion genügen:

$$U_k(n) = \begin{cases} T(n) & \text{falls } n \leq 2k \\ n/2 + U_k(n/2) + T(2k) & \text{sonst.} \end{cases}$$

  Zeigen Sie dies für Ihren Algorithmus.

- Zeigen Sie, dass $U_k(n) = n + O(T(2k)\log(n/k))$ ist. Für konstantes k erhält man also $U_k(n) = n + O(\log n)$.

(8 Punkte)

Abgabe der Hausaufgaben: Mittwoch, 15. 5. 2002, 10^{15} Uhr.