Technische Universität München
Institut für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr
Dr. Jens Ernst

WS 2003/2004
Übungsblatt 3
7. November 2003

Grundlegende Algorithmen

Letzter Abgabetermin: Freitag, 14. November 2003 (nach der Vorlesung)

Aufgabe 1 (10 Punkte)

Für Funktionen $e, f, g, h : \mathbf{N} \to \mathbf{R}^+$ bedeute $O(e) \circ O(f) := \{g \circ h : g \in O(e), h \in O(f)\}$, wobei $\circ \in \{+, *\}$. Analog für das Landau-Symbol o. Beweisen oder widerlegen Sie folgende Behauptungen:

a) $O(h) + O(g) \subseteq O(g + h)$ b) $O(h) * O(g) \subseteq O(g * h)$

c) $o(h) + o(g) \subseteq o(g + h)$ d) $o(h) * o(g) \subseteq o(g * h)$

Aufgabe 2 (5 Punkte)

Geben Sie eine Methode an, mit der man den Inhalt zweier Variabler a,b ohne Zuhilfenahme einer dritten Variablen vertauschen kann. Begründen Sie die Korrektheit.

Aufgabe 3 (10 Punkte)

In der Vorlesung wurde die Anzahl der Vergleiche untersucht, die *Insertion Sort* mit binärer Suche[1] beim Sortieren eines n-elementigen Arrays ausführt. Die Anzahl der Vergleiche ist aber nicht unbedingt ein realistisches Maß für die Zeitkomplexität des Algorithmus. Bestimmen Sie die Gößenordnung (in Landau-Notation) der *Gesamtzahl* an Operationen, die der Algorithmus für ein n-elementiges Array im worst-case ausführt. Geben Sie ein Beispiel für ein worst-case-Array an.
Hinweis: Beachten Sie hier insbesondere die Kosten für das zyklische Verschieben.

Programmieraufgabe 4 (15 Punkte)

Implementieren Sie den Algorithmus *Insertion Sort* mit binärer Suche[1] in einer Programmiersprache Ihrer Wahl. Die Verwendung von C wird empfohlen.
Das Programm soll eine Funktion der Form `void InsertionSort(int A[], unsigned n)` bereitstellen, die als Parameter ein zu sortierendes Array von ganzen Zahlen sowie dessen Länge entgegennimmt und die eigentliche Sortierung ausführt. Diese Funktion soll aufgerufen werden von einer `main()`-Funktion, welche sich die zu sortierenden Daten vom Benutzer eingeben läßt oder aus einem File liest. Das sortierte Array soll auf dem Bildschirm ausgegeben werden.

[1] siehe Vorlesung oder Buch von V. Heun, Kap. 2.1.3-2.1.4

Bitte geben Sie ihren *kommentierten* Programmcode in Papierform ab. Freiwillig können Sie ihn zudem per email an `griebsch@in.tum.de` schicken - schreiben Sie dann bitte `[GA: 3.4]` in die Betreff-Zeile.

Lesetip:

Zur Vorbereitung auf spätere Programmieraufgaben lesen Sie bitte die Kapitel 5.1 bis 5.9 aus dem Buch *The C Programming Language (2nd Edition)* von Brian W. Kernighan & Dennis Ritchie, Prentice Hall Verlag.