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

SS 2003

10. April 2003

Übungen zur Vorlesung
Effiziente Algorithmen

Blatt 1

Aufgabe 1: Seien f und g schließlich positive Funktionen, d.h. es gibt n_0 so dass $f(n) > 0$ für alle $n \geq n_0$. Beweisen oder widerlegen Sie die folgenden Behauptungen:

- $f(n) + g(n) = \Theta(\max(f(n), g(n)))$
- $(f(n) + g(n))^2 = O(f(n)^2) + O(g(n)^2)$
- Wenn $f(n) = O(g(n))$ ist, so auch $2^{f(n)} = O(2^{g(n)})$.
- $f(n) = \Theta(f(\frac{n}{2}))$

Aufgabe 2: Führen Sie eine detaillierte Analyse der Laufzeit, wie in der Vorlesung am Beispiel INSERTION-SORT vorgeführt, für die Routine MERGE, die beim Sortieren durch Mischen (MERGE-SORT) verwendet wird, durch.

Aufgabe 3: Beschreiben Sie einen Algorithmus zur binären Suche in einem sortierten Array in Pseudocode, und führen Sie eine detaillierte Analyse von dessen Laufzeit mit Hilfe der *Master*-Methode durch.

Aufgabe 4: Für welche der folgenden Rekursionsgleichungen kann das asymptotische Wachstum der Lösung $T(n)$ mit Hilfe der *Master*-Methode bestimmt werden? Geben Sie in den Fällen, wo dies möglich ist, möglichst scharfe asymptotische Schranken an.

- $T(n) = 8\,T(\frac{n}{2}) + 4n^3 + 2n^2$
- $T(n) = 9\,T(\frac{n}{3}) + n^2 \log n$
- $T(n) = 7\,T(\frac{3n}{8}) + 2n^2$
- $T(n) = 3\,T(\frac{3n}{5}) + n^2 \log n$