Technische Universität München
Institut für Informatik
Lehrstuhl für Effiziente Algorithmen
Volker Heun

SS 1999
Übungsblatt 2
18. Mai 1999

Grundlegende Algorithmen

Abgabetermin: 1. Juni

Wegen der Pfingstferien fallen am 25./26. Mai die Vorlesungen und die Übungen aus!

Am Mittwoch, dem 18. Mai, finden die Übungen von 15^{15}–16^{45} ausnahmsweise in S2229 statt!

Aufgabe 1

Für ein Problem gibt es zur Lösung die Algorithmen $\mathcal{A}$ und $\mathcal{B}$. Die Zeitkomplexität ist durch durch folgenden Funktionen gegebenen: $T_{\mathcal{A}}(n) = 1024 * n \log(n)$ und $T_{\mathcal{B}}(n) = 32 * n^{3/2}$. Welchen Algorithmus würden Sie wählen und warum?

Aufgabe 2

Vergleichen Sie das Wachstumsverhalten der folgenden Funktionen (von $\mathbb{N} \to \mathbb{R}_+$) mit Hilfe der Landauschen Symbole o, O, Θ:

1)	$n \log(n)$	2)	2^n
3)	$n^2 \log(n)$	4)	$n^3 \lvert \cos(n\pi/2) \rvert$
5)	n	6)	$n^{3/4}(\log(n))^4$
7)	$n \operatorname{loglog}(n)$	8)	$n \log^2(n)$
9)	3^n	10)	$n \cdot \sqrt{n}$

Hinweis: Nutzen Sie (sofern möglich) die Transitivität der Landauschen Symbole o, O, Θ aus.

Aufgabe 3

Versuchen Sie die folgenden Ausdrücke mit Hilfe der Θ-Notation möglichst einfach darzustellen.

1. $\sum_{i=1}^{n} \frac{1}{i}$

2. $\sum_{i=1}^{n} \frac{1}{i^2}$
3. $\log(n!)$
4. $(\log(n))^{\frac{\log(n)}{\log(\log(n))}}$
5. $8^{\lg\lg(n)}$

Z.B. ist $2^{\log\log(n)+5} = \Theta(\log(n))$.

Aufgabe 4

a) Der folgende Algorithmus fügt in ein sortiertes n-elementiges Teilfeld ein neues Element ein, so daß das neue Teilfeld der Größe $n+1$ wiederum sortiert ist.

```
Insert(int new_elt, int array[], int n)
{
  for (int j=0; j<n && array[j]<new_elt; j++) ;
  for (int k=n; k>j; k--) array[k]=array[k-1];
  array[j]=new_elt;
}
```

Beachten Sie, daß das Feld `array` die Größe $m > n$ hat, aber daß sich nur in den ersten n Position die sortierten Elemente befinden.
Konstruieren Sie mit Hilfe dieser Routine eine Algorithmus zum Sortieren von von Feldern.

b) Analysieren Sie die Komplexität (Anzahl von Vergleichen) Ihres Sortieralgorithmus aus Teil a).

c) Geben Sie einen besseren Algorithmus für das Problem in Teil a) an und geben Sie die hierfür maximal benötigte Anzahl von Vergleichen an.
Hinweis: Es geht mit $O(\log(n))$ Vergleichen.

d) Konstruieren Sie mit Hilfe Ihres Einfüge-Algorithmus aus Teil c) einen Sortieralgorithmus und analysieren Sie dessen Komplexität (Anzahl benötigter Vergleiche).

e) Vergleichen Sie die benötigten Anzahl von Vergleichen von beiden Algorithmen und überlegen Sie, ob in Praxis wirklich einer der beiden Algorithmen wesentlich besser ist.

f) Macht es einen Unterschied, wenn man Listen anstelle von Feldern verwendet?