

Übung zur Vorlesung Datenstrukturen und Algorithmen

Aufgabe T1

Ordnen Sie die folgenden Funktionen in der Reihenfolge ihres asymptotischen Wachstums (ohne lange Nachzudenken). Mit $\log n$ bezeichnen wir den Logarithmus zur Basis 2 und $e = 2.714\dots$ ist die Eulerkonstante.

a) $\log(n)$

b) $\log(n^3)/\log \log(n)$

c) $n^{\log \log n}$

d) $\log(n)^{\log n}$

e) n^e

f) $n \log n$

g) 2^n

h) $n^{\sqrt{n}}$

Aufgabe T2

Beweisen Sie: $\frac{1}{n+1} = \frac{1}{n} + O\left(\frac{1}{n^2}\right)$

Aufgabe T3

Finden Sie eine Funktion $f: \mathbf{N} \rightarrow \mathbf{R}$, so daß folgendes gilt:

$$\sum_{k=1}^n \frac{1}{k} = f(n) + O(1)$$

Aufgabe T4

Entwerfen Sie einen einfachen Algorithmus, der zwei doppelte verkettete Listen aneinanderhängt und auf diese Weise eine neue doppelt verkettete Liste erzeugt. Können Sie dabei auf bedingte Verzweigungen verzichten?

Schreiben Sie den Algorithmus in Pseudocode oder in einer Programmiersprache nieder. Sie erhalten als Eingabe die Köpfe beider Listen.

Aufgabe H1 (1+1+1+1+2+2 Punkte)

Schätzen Sie folgende Ausdrücke durch möglichst einfache Funktionen ab, wobei ein additiver Fehler von $O(n)$ erlaubt ist. Beispielsweise gilt $2n(n-1) = 2n^2 + O(n)$. Ein Beweis ist nicht notwendig.

a) $n^2(1 + 1/n)$

b) $\sum_{k=1}^n k$

c) $n!/n^n$

d) $\log(\cos(en)n + n^5)$

e) $(n+1)^{n+1}/n^n$

f) $\binom{n}{3}$

Aufgabe H2 (8 Punkte)

Beweisen Sie formal mithilfe der Definition der O -Notation, daß

$$\sqrt{2n(n+2)(n-1)} = \Theta(n^{3/2}).$$

Aufgabe H3 (10 Punkte)

Entwerfen Sie einen Algorithmus oder schreiben Sie ein Programmfragment, welches die Reihenfolge zweier aufeinanderfolgenden Knoten in einer doppelt verketteten Liste vertauscht. Als Eingabe erhalten Sie den ersten der beiden Listenknoten. Vertauschen Sie aber nicht den Inhalt der Knoten, sondern verwenden sie „Zeigerverbiegen“.

Wie in der Vorlesung gezeigt, sind Listenknoten so implementiert (ohne die Methoden):

```
class Listnode<K, D> {
    K key;
    D data;
    Listnode<K, D> pred, succ;
}
```

Eine falsche Lösung, die aber eine syntaktisch korrekte Javamethode ist, könnte so aussehen:

```
void vertauscheMitDemNachfolger(Listnode<K, D> x) {
    x.succ.pred = x.pred;
    x.succ.succ = x.succ;
    x.succ = x.succ.succ;
    x.pred = x.succ.pred;
}
```

Ihre Lösung könnte ähnlich aussehen, muß aber das richtige tun.