

## Übungen zur Vorlesung Informatik I

Blatt 13

**Im allgemeinen wird zur Klausur zugelassen, wer 120 Punkte aus den Übungsblättern 1-13 erreicht hat. Somit enthält dieses Übungsblatt nur Bonuspunkte.**

**Abgabe bis spätestens Donnerstag, 5.2.04, 11:00 Uhr im Briefkasten in der Theresienstr.!**

### **Schriftliche Aufgabe S-50:**

**4 Punkte**

Betrachten Sie die naive Implementierung der Funktion `rev`, die eine Liste umdreht

```
let rec rev_naive = function
  [] -> [] |
  h::t -> (rev_naive t)@[h]
```

sowie die clevere Variante

```
let rev_clever =
  let rec rc acc = function
    [] -> acc |
    h::t -> rc (h::acc) t
  in rc []
```

Als Größe  $n$  der Eingabe an diese Funktionen definieren wir – wie üblich – die Länge der Liste.

Geben Sie zu den beiden Funktionen jeweils die Laufzeit in der Form  $O(f(n))$  an. Begründen Sie Ihre Antworten.

### **Schriftliche Aufgabe S-51:**

**6 Punkte**

Betrachten Sie die folgende (leicht modifizierte Version der bekannten) Funktion `ispath : 'a list * 'a bintree -> bool`, welche zu einer Liste und einem Baum entscheidet, ob die Liste als Pfad in dem Baum vorkommt.

```
let rec ispath l =
  let rec ispathaux b = match l with
    [] -> (isempty b) |
    h::t -> h = root b && (ispath t (left b) || ispath t (right b))
  in ispathaux
```

Im folgenden interessieren wir uns nur für die Funktion `ispathaux : 'a bintree -> bool`.

- a) Definieren Sie sinnvoll die Größe der Eingabe.
- b) Wovon – ausser von der Größe  $n$  der Eingabe – hängt die Laufzeit der Funktion `ispathaux` noch ab? Geben Sie jeweils an, wann der *best case* und der *worst case* vorliegt.
- c) Geben Sie jeweils die Größenordnung der Laufzeit von `ispath` im *best case* und in *worst case* an. Begründen Sie Ihre Antworten.

**Schriftliche Aufgabe S-52:**

**5 Punkte**

Beweisen Sie die folgenden Aussagen:

- a)  $\log n! = O(n \cdot \log n)$
- b)  $n \cdot \log n = O(\log n!)$
- c)  $n \cdot \log n = O(n^{1+\epsilon})$  für jedes  $\epsilon > 0$

**Schriftliche Aufgabe S-53:**

**5 Punkte**

Beweisen Sie die folgenden Eigenschaften der  $O()$ -Notation:

- a) Für jedes  $f$  ist  $f(n) = O(f(n))$ .
- b) Ist  $f(n) = O(g(n))$  und  $g(n) = O(h(n))$ , dann auch  $f(n) = O(h(n))$ .
- c)  $f(n) + g(n) = O(\max(f(n), g(n)))$ .