

Proinformatik

Marco Block

Donnerstag, den 23. Juli 2009

Die Aufgaben mit einer *ungeraden* Nummer sollen Sie während des Tutoriums bearbeiten, die anderen Aufgaben (hand?-)schriftlich zu Hause. Sehen Sie das als eine Übung für die Klausur an und verzichten Sie darauf Google, Hugs o.ä. zu benutzen. Geben Sie Montag ihre Lösung bei einem der Tutoren ab.

1. **Wiederholung** Diese Aufgaben sollten nicht schwer für Sie sein

- Schreiben Sie eine Funktion `nth`, die das n-te Element einer Liste zurückgibt.
- Schreiben Sie eine Funktion `slice`, die ein Stück aus einer Liste extrahiert:

```
slice 3 7 [0..10] = [3,4,5,6,7]
```

- Schreiben Sie eine Funktion `teiler`, die für eine natürliche Zahl eine Liste mit ihren Teilern zurückgibt.

```
teiler 12 = [1,2,3,4,6,12]
```

- Schreiben Sie eine Funktion `pack`, die eine Liste komprimiert, indem sie aufeinanderfolgende Duplikate eines Elements durch ein Tupel aus dem Element und der Zahl der Wiederholungen ersetzt. *Bonuspunkte* Schreiben Sie `pack` als Faltung. *Bonusbonuspunkte* Schreiben Sie `pack` als Entfaltung.

```
pack [1,1,2,4,4,5] = [(1,2), (2,1), (4,2), (5,1)]
```

Hinweis: Benutzen sie `length` und `gruppieren`.

- Schreiben Sie eine Funktion `unpack`, die die Umkehrfunktion von `pack` ist. *Bonuspunkte* Schreiben Sie `unpack` als Entfaltung. *Bonusbonuspunkte* Schreiben Sie `unpack` als Faltung.

2. **Falten** Sie haben in der Vorlesung und auf den Übungszetteln schon Insertion- und Selectionsort kennengelernt. Definieren sie Insertionsort mit einer Faltung und Selectionsort mit einer Entfaltung. Wenn Sie sich unterfordert fühlen, können Sie auch BubbleSort mit einer Entfaltung angeben.

3. **Module** Schreiben Sie sich ein Modul `Vorlesung`, dass z.B. die Funktionen aus der heutigen Vorlesung enthält. Schreiben Sie sich ein zweites Modul `Test`, in dem Sie das Modul `Vorlesung` mit geeigneten Testfunktionen ausprobieren. Spielen Sie mit den Anweisungen zur Sichtbarkeit herum, um Funktionen im Modul `Vorlesung` zu verstecken oder nur bestimmte Funktionen von dort zu importieren.

4. **Ein bisschen Laufzeit** Überlegen Sie sich, wieviele Vergleiche Insertionsort und Selectionsort in Abhängigkeit von der Länge der Liste haben. Machen Sie sich ggf. ein paar kleine Beispiele.

5. **Mergesort** In der Vorlesung haben Sie Mergesort kennengelernt. Dabei wurde das Verfahren so vorgestellt, dass es die Eingabeliste zuerst rekursiv halbiert, bis nur noch einzelne Elemente in den Teillisten enthalten sind. Danach wurden die Teillisten mit **merge** wieder zu sortierten Listen zusammengefügt.

Können Sie eine Variante angeben, die „von unten nach oben“ arbeitet? Sie soll mit den einzelnen Elementen anfangen und nach und nach immer größere sortierte Teillisten bauen. Welche Variante ist ihrer Meinung nach effizienter? Warum?