



Übungen zur „Praktischen Informatik III“, WS 2008/09

Prof. Dr. R. Loogen · Fachbereich Mathematik und Informatik · Hans-Meerwein-Straße, D-35032 Marburg

Nr. 7, Abgabe: 3. Dezember 2008 vor der Vorlesung

15. Zeilennummern

3 Punkte

Schreiben Sie eine Funktion `addLineNumbers :: String -> String`, die einen Text um Zeilennummern erweitert. Verwenden Sie die vordefinierten Funktionen `lines`, `unlines`, `zip` und `map`.

16. Buchstabenhäufigkeit

5 Punkte

- (a) Beim Mischsortieren wird die zu sortierende Folge in zwei Teilfolgen zerlegt, die rekursiv mit demselben Verfahren sortiert werden. Die zwei sortierten Teilfolgen werden dann zu einer sortierten Gesamtfolge gemischt.

Schreiben Sie `mergeSort :: ([t] -> [t] -> [t]) -> [t] -> [t]` zum Mischsortieren, wobei die Mischfunktion als Argument übergeben wird. / 1

- (b) Definieren Sie unter Verwendung von `mergeSort` eine Funktion / 2

`sortfold :: [(Char, Int)] -> [(Char, Int)]`

die eine Liste von Paaren aus je einem Buchstaben und einer Zahl nach den Buchstaben sortiert und dabei Paare mit gleichen Buchstaben durch die Addition der Zahlen zusammenfasst.

Beispiel: `sortfold [('c',2), ('d',1), ('a',4), ('d',2), ('c',5)]`
`==> [('a',4), ('c',7), ('d',3)]`

- (c) Implementieren Sie unter Verwendung der in (a) und (b) definierten Funktionen die Funktion `frequency :: [Char] -> [(Char, Int)]`, die zu einer gegebenen Zeichenkette eine Liste aller darin vorkommenden Zeichen zusammen mit der Häufigkeit ihres Auftretens erstellt. Die Ergebnisliste soll dabei nach der Häufigkeit sortiert sein. / 2

Beispiel: `frequency "battat" ==> [('b',1), ('a',2), ('t',3)]`

17. Modellierung natürlicher Zahlen

4 Punkte

Gegeben seien die folgenden Definitionen zur Modellierung natürlicher Zahlen in Haskell:

```
data Nat = Zero | Succ Nat deriving Show
foldN :: a -> (a->a) -> Nat -> a
foldN z s Zero      = z
foldN z s (Succ n) = s (foldN z s n)
```

- (a) Definieren Sie *als Instanz von* `foldN` die binären Operationen / 2

`addN`, `mulN`, `powerN :: Nat -> Nat -> Nat`
 zur Addition, Multiplikation und Exponentiation natürlicher Zahlen.

- (b) Definieren Sie *als Instanz von* `data Maybe a = Nothing | Just a` / 2

`foldN` die nebenstehend durch Pattern Matching definierte Vorgängerkfunktion auf natürlichen Zahlen.

```
subN :: Nat -> Maybe Nat
subN Zero      = Nothing
subN (Succ n) = Just n
```