

---

SoSe 2014  
Prof. Dr. Margarita Esponda  
ProInformatik II: Funktionale Programmierung  
**7. Übungsblatt** (8. Tag)

---

### 1. Aufgabe

Analysiere die Komplexität der Mergesort-Algorithmen aus der Vorlesung.

### 2. Aufgabe

Ein Element einer Liste von  $n$  Objekten stellt die absolute Mehrheit der Liste dar, wenn das Element mindestens  $\left(\frac{n}{2} + 1\right)$  mal in der Liste vorkommt. Definieren Sie

eine **majority**-Funktion, die mit linearem Aufwand das Majority-Element der Liste findet, wenn eines existiert oder sonst **Nothing** zurückgibt.

Die Signatur der Funktion soll wie folgt aussehen:

**majority** :: (Eq a) => [a] -> **Maybe** a

### 3. Aufgabe

a) Definieren Sie folgende Funktionen für den algebraischen Datentyp **Nat** aus der Vorlesung.

**nungerade** :: Nat -> Bool

**cutSub** :: Nat -> Nat -> Nat    --  $a - b \Rightarrow 0$ , wenn  $b \geq a$

**nmin** :: Nat -> Nat -> Nat

b) Programmieren Sie folgende Hilfsfunktionen, die das Testen der Funktionen vereinfachen sollen.

**nat2Int** :: Nat -> Integer

**int2Nat** :: Integer -> Nat

Anwendungsbeispiel: **nat2Int** (S (S (S Zero))) => 3

**int2Nat** 5 => (S (S (S (S (S Zero)))))

c) Definieren Sie eine **zsimplify** Funktion, die eine beliebige **ZInt** Zahl in seine normal Form umwandelt. D.h. (Z a Zero) oder (Z Zero a).

d) Programmieren Sie folgende Funktionen für den algebraischen Datentyp **ZInt** aus der Vorlesung.

**zneg** :: ZInt -> ZInt    -- negative Zahl

**zpow** :: ZInt -> Nat -> ZInt    -- Potenz-Funktion

**zabs** :: ZInt -> ZInt    -- Absoluter Wert

**zggt** :: ZInt -> ZInt -> ZInt    -- Größter gemeinsamer Teiler

e) Definieren Sie folgende Hilfsfunktionen.

**zint2Int** :: ZInt -> Integer    -- transformiert von ZInt nach Integer

**int2ZInt** :: Integer -> ZInt    -- transformiert von Integer nach ZInt