

**Tentamen Functioneel Programmeren (211205)**

22 januari 2013  
13.45 – 17.15 uur

Opmerkingen vooraf:

- U mag het dictaat *Functional Programming – an overview* bij dit tentamen gebruiken, verder niets.
- Voor Haskell: u mag alleen gebruik maken van die voorgedefinieerde Haskell functies die equivalent zijn aan de built-in Amanda functies (beschikbaar op lijst).
- **Geef bij elke functie die u definieert het type.**
- Beoordeling: er zijn drie opgaven, de zwaarte is bij elke opgave aangegeven.
- De elegantie van de oplossing zal ook een rol spelen, dus gebruik geen onnodige hulpfuncties.
- Succes!

**Opgave 1 (25 punten).**

**a.** Schrijf een functie `klein` die uit een gegeven lijst van getallen alle getallen selecteert, die kleiner zijn dan het gemiddelde van alle getallen in de gegeven lijst. Bijvoorbeeld: `klein [4,3,6,3,5] = [4,3,3]`.

**b.** Een lijst van getallen is dalend, als elk volgend element in de lijst kleiner is dan het voorgaande. Bijvoorbeeld, de lijst `[9,7,5,4,2]` is dalend, terwijl de lijst `[8,7,5,6,6,2]` dat niet is. Schrijf een functie `dalend`, die bepaalt of een lijst dalend is of niet.

**b.** Een lijst is zwak dalend, als elk volgend element in de lijst kleiner is dan het gemiddelde van alle voorafgaande elementen in de lijst. Bijvoorbeeld, bovengenoemde lijst `[8,7,5,6,6,2]` is wel zwak dalend. Schrijf een functie `zwakDalend`, die bepaalt of een lijst zwak dalend is of niet.

**Opgave 2 (35 punten).** Gegeven is het type

```
tree ::= Leaf | Node num tree tree
```

- a. Schrijf een functie `count` die telt hoe vaak een gegeven getal in een gegeven boom voorkomt.
- b. Schrijf een functie `zoekMax1` die van een gegeven boom het grootste getal oplevert (dat grootste getal kan eventueel vaker dan één keer in de boom voorkomen).

Een boom van type `tree` is gesorteerd, als voor elke knoop in de boom geldt dat alle getallen in de linker subboom bij die knoop kleiner zijn dan het getal aan die knoop zelf, en alle getallen in de rechter subboom groter (dus alle getallen in een gesorteerde boom zijn verschillend).

- c. Schrijf een functie `sorted` die test of een boom gesorteerd is of niet.
- d. Schrijf een functie `add` die een getal aan een gesorteerde boom toevoegt, zodanig dat de resulterende boom ook weer gesorteerd is.
- e. Schrijf een functie `voegsamem` die twee gesorteerde boom samenvoegt tot een nieuwe boom die ook weer gesorteerd moet zijn.
- f. Schrijf een functie `groterDanGem` die uit een gesorteerde boom alle getallen weggooit die kleiner dan of gelijk aan het gemiddelde zijn. De resulterende boom moet ook weer gesorteerd zijn.
- g. Schrijf een functie `zoekMax2` die van een gegeven gesorteerde boom het grootste getal oplevert (dus als bij b, alleen is nu de boom gesorteerd).

**Opgave 3 (40 punten).** Een gerichte graaf is een verzameling van nodes (punten) en edges (bindingen). Laat de nodes aangegeven zijn door nummers, en de edges door geordende paren van de nummers van de betreffende nodes.

- a. Definieer een type `graaf` waarin dergelijke gerichte grafen kunnen worden ondergebracht.
- b. Schrijf een functie `bereikbaar1` die, gegeven een gerichte graaf en een startnode, de lijst van alle nodes oplevert die vanuit de startnode in één stap bereikbaar zijn.
- c. Schrijf een functie `paden` die, gegeven een gerichte graaf en een begin- en eindnode, alle paden oplevert van de begin- naar de eindnode. Een pad bestaat uit edges die alleen in de goede richting mogen worden doorlopen.

- d.** Schrijf een functie `subgraaf` die, gegeven een graaf en een node, de samenhangende subgraaf oplevert waar deze node onderdeel van uitmaakt. Hierbij is de richting van de edges niet van belang.
- e.** Schrijf een functie `componenten` die het aantal samenhangende subgrafen bepaalt waar een gegeven graaf uit bestaat.
- f.** Een edge is een *brug* als na verwijdering van die edge de graaf één samenhangende subgraaf meer bevat. Schrijf een functie `bevatBrug` die test of een graaf een brug bevat.