

FACULTEIT EWI  
Datum: 17 juni 2008

**Tentamen Functioneel Programmeren (211205)**

26 juni 2008  
9:00 – 12:30 uur

Opmerkingen vooraf:

- u mag het dictaat bij dit tentamen gebruiken,
- het tentamen bestaat uit drie opgaven, die alle even zwaar wegen,
- geef bij *elke* functie die u definieert het type,
- succes!

**Opgave 1.** In deze opgave moet u een aantal operaties voor verzamelingen schrijven. In een verzameling komt een object niet dubbel voor, en een verzameling is ongeordend.

- a. Definieer een type voor verzamelingen.
- b. Schrijf een functie `subset` die test of een gegeven verzameling deelverzameling is van een andere verzameling.
- c. Schrijf een functie `union` die de vereniging van twee verzamelingen bepaalt.
- d. Twee verzamelingen zijn disjunct als ze geen elementen gemeenschappelijk hebben. Schrijf een functie `disjoint` die test of twee verzamelingen disjunct zijn.
- e. De machtsverzameling van een verzameling  $A$  is de verzameling van alle deelverzamelingen van  $A$ . Schrijf een functie `powerset` die de machtsverzameling van een verzameling oplevert.

**Opgave 2.** Binaire bomen waarin grote databestanden gesorteerd opgeslagen moeten worden bevatten veelal alleen gegevens aan de interne knopen (de nodes) en niet aan de bladeren. Een mogelijke manier om in dergelijke toepassingen tóch gegevens aan de bladeren te kunnen opslaan is met het volgende type voor binaire bomen:

```
boom ::= Single num
      | Double num num
      | Branch num boom boom
```

Een boom van dit type heeft getallen op zowel de bladeren als de interne knopen; er zijn bladeren met *één* getal en bladeren met *twee* getallen.

Een boom van de vorm `Branch n t1 t2` is *gesorteerd* als geldt dat alle getallen in `t1` kleiner zijn dan `n` of gelijk aan `n`, alle getallen in `t2` groter zijn dan `n`, en de twee subbomen `t1` en `t2` ook gesorteerd zijn.

Een boom van de vorm `Double a b` is gesorteerd als  $a \leq b$ ; een boom van de vorm `Single a` is gesorteerd.

- a. Schrijf een functie `insert` die een getal toevoegt aan een boom. Er moet gelden dat wanneer de gegeven boom gesorteerd is, de opgeleverde boom ook gesorteerd is.
- b. Schrijf een functie `maakboom`, die, gegeven een niet lege lijst getallen een gesorteerde boom oplevert die deze getallen bevat. U mag de functie `insert` gebruiken, ook als u die niet heeft kunnen schrijven.
- c. Schrijf een functie `boomNaarLijst` die, gegeven een boom, een lijst oplevert die alle getallen uit de boom bevat. Er moet gelden dat wanneer de boom gesorteerd is, de opgeleverde lijst ook gesorteerd is. U mag geen hulpfunctie `sort` gebruiken.
- d. Schrijf een functie `boomsort` die gegeven een lijst getallen een gesorteerde lijst met deze getallen oplevert. Deze functie moet de getallen uit de lijst eerst in een boom zetten, en ze er daarna weer uithalen. U mag de functies uit de voorgaande delen van de opgave gebruiken, ook als u die niet heeft kunnen schrijven.
- e. Generaliseer bovenstaand type zodat er aan interne nodes en aan de bladeren waardes van een willekeurig type opgeslagen kunnen worden. Bij elke knoop en elk blad moeten waarden van hetzelfde type staan.

**Opgave 3.** Gegeven is het type van gerichte graaf:

```
node == num
edge == (node,node)
graph == ([nodes], [edge])
```

Nodes zijn dus gewoon getallen, u mag aannemen dat alle nodes in een graaf verschillende getallen zijn.

**a.** De *ingraad* van een node is het aantal binnenkomende edges van die node. Schrijft een functie *ingraad* die de ingraad van een node in een graaf berekent. U moet deze twee functies op drie manieren definiëren: met recursie, met lijstcomprehensie, en met hogere orde functies.

Let wel: u mag geen hulpfuncties en geen *where*-clauses gebruiken.

**b.** Een edge in een graaf is een *brug* in die graaf als de graaf na weghalen van die edge in meerdere subgrafen uiteenvalt. Dat wil zeggen, dat er na weghalen geen edges meer tussen deze subgrafen bestaan. Schrijf een functie *isbrug* die nagaat of een edge een brug is in een gegeven graaf.

Opmerking: bij deze opgave is het gericht zijn van de graaf niet van belang. En: we nemen aan dat de graaf voordat de brug wordt weggehaald, samenhangend is, dus dat alle nodes door een pad (mogelijk van lengte groter an één) met elkaar verbonden zijn.