Fakultät IV Elektrotechnik und Informatik
MPGI1 WS 2008/09Pepper / Kleeblatt / Tutoren

Test 3

(Permutation: 3-3-3)

Datum: 26./27.01.2008

Aufgabe	*1*	*2*	*3*	*Σ*
max. Punkte	*2*	*2*	*2*	*6*
erreichte Punkte				

Aufgabe 1 - Aufwand (O-Kalkül) (2 Punkte):

Bestimmen Sie die Aufwandsklasse für die Laufzeit der Funktion `f`. Leiten Sie zunächst die Rekurrenzrelation her. Für Operationen mit konstantem Aufwand können Sie dabei beliebige Konstanten annehmen. Geben Sie dann an, welche Zeile der Tabelle Sie verwenden und wie die Variablen in der entsprechenden Zeile belegt werden müssen.

	Rekurrenzrelation		$A \in$
1	$A(n) = A(n-1) + bn^k$		$\mathcal{O}(n^{k+1})$
2	$A(n) = cA(n-1) + bn^k$	mit $c > 1$	$\mathcal{O}(c^n)$
3	$A(n) = cA(n/d) + bn^k$	mit $c > d^k$	$\mathcal{O}(n^{\log_d c})$
4	$A(n) = cA(n/d) + bn^k$	mit $c < d^k$	$\mathcal{O}(n^k)$
5	$A(n) = cA(n/d) + bn^k$	mit $c = d^k$	$\mathcal{O}(n^k \log_d n)$

```
FUN f : nat -> nat
DEF f(0) == 1
DEF f(n) == LET
                z == h(n)
             IN
                2*f(n/2) + z

FUN h: nat -> nat
DEF h(0) == 0
DEF h(n) == n * (n - 1)
```

Name: ..... **Matr.Nr:** ..... **Tut.Nr:** .....

Aufgabe 2 - Listenfunktionale (2 Punkte):

Deklarieren und definieren Sie eine Funktion `equal?`, der zwei ganzzahlige Listen übergeben werden. Die Funktion soll überprüfen, ob beide Listen gleich sind. Sie können davon ausgehen, dass die Länge der Listen gleich ist. Prüfen Sie erst die paarweise Gleichheit der Elemente. Bilden Sie dann die Konjunktion der Ergebnisse.
Verwenden Sie keine direkte Rekursion sondern *Listenfunktionale*!

Beispiel: `equal?(-(1) :: 2 :: <>, -(1) :: 1 :: <>)  =  false`

Aufgabe 3 - Sortieren (2 Punkte):

Gegeben ist die Sortierfunktion `sort` und die Deklaration ihrer Hilfsfunktion `f`. Definieren Sie `f` und benennen Sie das hier benutzte Sortierverfahren.

```
FUN sort: seq[int] -> seq[int]
DEF sort(<>) == <>
DEF sort(s :: <>) == s :: <>
DEF sort(S) == LET  i               == #(S)/2
                    (Links, Rechts) == split(i, S)
               IN   f(sort(Links), sort(Rechts))


FUN f : seq[int] ** seq[int] -> seq[int]
```

Name: ..... **Matr.Nr:** ..... **Tut.Nr:** .....