Theoretische Informatik

Prof. Dr. B. Nebel, Prof. Dr. G. Lausen
M. Ragni, K. Simon, and C.-N. Ziegler
Wintersemester 04/05

Universität Freiburg
Institut für Informatik

Gruppenübung

Wiederholung
Formale Sprachen, Probleme, Entscheidungsprobleme, Berechenbarkeitsbegriff, Registermaschine.

Aufgabe 1 (Random-Access-Maschinen)
Multiplikation.
Beim Start der RAM sind die Registerinhalte $< R0 >= x$, $< R1 >= y$ und $< Ri >= 0$ für $i \geq 2$ vorgegeben. Geben Sie einen Algorithmus an, so daß die RAM nach endlich vielen Schritten stoppt, wobei beim Stopp $< R0 >= 2 \cdot x \cdot y$ gelten soll.

Potenzierung.
Beim Start der RAM sind die Registerinhalte $< R0 >= x$, $< R1 >= y$ und $< Ri >= 0$ für $i \geq 2$ vorgegeben. Geben Sie einen Algorithmus an, so daß die RAM nach endlich vielen Schritten stoppt, wobei beim Stopp $< R0 >= x^y$ gelten soll.

Aufgabe 2 (Problemformalisierung)

1. Formalisieren Sie die folgenden Probleme:
   (a) Prim, d.h. ob eine gegebene natürliche Zahl eine Primzahl ist.
   (b) Tree, d.h. ob ein gegebener Graph ein Baum ist.
   (c) Rucksack, d.h. ob es zu einer Menge von Objekten M (mit Gewicht) und einem Maximalgewicht b eine Teilmenge von M gibt, s. d. die Summe der Gewichte der Objekte gleich b ist.
   (d) Partition, d.h. ob sich eine Menge von natürlichen Zahlen so teilen lässt, dass die Summe beider Teilmengen identisch ist.
   (e) TSP, d.h. ob zu einer Menge von Distanzen zwischen mehreren Städten und einer Maximaldistanz eine Rundreise existiert, die kürzer als die Maximaldistanz ist.
2. Skizzieren Sie Algorithmen (Pseudocode), die diese Probleme lösen.

Aufgabe 3 (Laufzeitverhalten)
Bestimmen Sie das Laufzeitverhalten der folgenden Funktion:

```
public static int method (int n){
       int x = 0;
       int y = 0;

       for (int i = 1; i < n; i++){
           for (int j = i; j < n; j++){
                x = x + 1;
           }
           for (j = 1; j < i; j++){
                for (k = 1; k < i; k++){
                     y = y + 1;
                }
           }
        }
      return x + y;
 }
```

Aufgabe 4 (*O*- Notation)
Beweisen oder widerlegen Sie:

1. $n^2 = O(n)$
2. $n = O(n^2)$
3. $n^n = O(n^2)$
4. $n^{log(n^2)} = O(n^2)$
5. $2^n = O(n^{12})$