Prof. Dr. Matthias Rarey
M. v. Behren, F. Heitmann, M. Hilbig,
F. Lauck, T. Otto, J. Röwekamp

Zentrum für Bioinformatik Hamburg (ZBH)
Abteilung für Algorithmisches Molekulares Design

Wintersemester 2012/2013

Übung zur Vorlesung Algorithmen und Datenstrukturen
Übung 2

Abgabe: 05.11.2012, 12 Uhr

Aufgabe 1: Rekurrenzgleichungen und Rekursionsbäume

a)

Gegeben sei folgender Algorithmus (dessen genaue Funktionsweise nachfolgend nicht wichtig ist):

```
FUN(A)
if A.länge < 3
    return 5
else
    x = A.länge/3
    y = FUN(A[1 .. x])
    z = FUN(A[2 · x + 1 .. A.länge])
    r = 0
    for i = 1 to A.länge
        for j = i + 1 to A.länge
            r = r + A[i] · A[j]
return r
```

Leiten Sie eine Rekurrenzgleichung für die Laufzeit der Methode FUN in Abhängigkeit von der Arraygröße ab. Begründen Sie Ihre Gleichung. (2 Punkte)

b)

Sei

$$T(n) := \begin{cases} c, & \text{für } n \leq 1 \\ 16 \cdot T(\frac{n}{4}) + n^2, & \text{sonst} \end{cases}$$

eine Rekurrenzgleichung (c ist dabei eine Konstante).

Bestimmen Sie die Größenordnung der Funktion $T : \mathbb{Q}^+ \to \mathbb{Q}^+$ und zwar (a) mittels der Substitutionsmethode und (b) mittels des Mastertheorems. Ihre Ergebnisse sollten zumindest hinsichtlich der O-Notation gleich sein, so dass Sie etwaige Rechenfehler entdecken können! Beweisen Sie bei (a) auch die Korrektheit Ihrer Abwicklung mittels Induktion für alle n mit $n \mod 4 = 0$. (Hinweis: Führen Sie den Induktionsschritt von n nach 4n durch). (6 Punkte)

c)

Sei

$$T(n) := \begin{cases} c, & \text{für } n \leq 1 \\ a \cdot T(\frac{n}{b}) + f(n), & \text{sonst} \end{cases}$$

eine Rekurrenzgleichung.

1. Beweisen Sie, dass die Tiefe des Rekursionsbaumes $t = \log_b n$ $(= \frac{\log n}{\log b})$ beträgt.
2. Geben Sie eine Formel für die Anzahl der Blätter abhängig von n, a, b, c und $f(n)$ an (wobei nicht alle Variablen benötigt werden).

(2 Punkte)

Aufgabe 2: Sortieren und ein Korrektheitsbeweis

1. Die Laufzeit von INSERTION-SORT ist in $\Theta(n^2)$, die von MERGE-SORT in $\Theta(n \cdot \log n)$. Dennoch gibt es Fälle, in denen INSERTION-SORT schneller ist als MERGE-SORT. Geben Sie eine Regel an, wie ein Array der Länge n mit Schlüsselwerten belegt werden kann, so dass das Array mit INSERTION-SORT schneller sortiert wird als mit MERGE-SORT. Begründen Sie Ihre Regel, in dem Sie die Laufzeit der Sortierverfahren für die entsprechenden Eingabe-Arrays abschätzen. (2 Punkte)
2. Im Buch von Cormen et al. finden Sie (zu Beginn von Kapitel 2) den Korrektheitsbeweis für den Algorithmus INSERTION-SORT. Im Beweis wird eine Schleifeninvariante aufgestellt und zum Beweis der Korrektheit genutzt. Beweisen Sie formal und in analoger Weise die Korrektheit von SELECTION-SORT aus der Vorlesung (Kapitel 3, Folie 4). (4 Punkte)