Albert-Ludwigs-Universität Freiburg | PD Dr. Cyrill Stachniss
Institut für Informatik
Einführung in die Informatik
Sommersemester 2013 | Dr. Rainer Kümmerle

Übungsblatt 10

Abgabe bis Freitag, 05.07.2013, 10:00 Uhr

Hinweis:

Aufgaben immer per E-Mail (eine E-Mail pro Blatt und Gruppe) an den zuständigen Tutor schicken (Bei Programmieraufgaben Java Quellcode und evtl. benötigte Datendateien).

Aufgabe 10.1

Jede symmetrische positiv definite Matrix $A \in \mathbb{R}^{n \times n}$ kann eindeutig in der Form

$$A = LL^T$$

geschrieben werden. Dabei ist L eine untere Dreiecksmatrix und kann mittels Cholesky-Zerlegung bestimmt werden [1]. Die Elemente der Dreiecksmatrix L lassen sich wie folgt bestimmen:

$$\begin{aligned} L_{ii} &= \sqrt{a_{ii} - \sum_{k=0}^{i-1} L_{ik}^2} \\ L_{ji} &= \frac{1}{L_{ii}} \left(a_{ij} - \sum_{k=0}^{i-1} L_{ik} L_{jk} \right) \qquad j = i+1, i+2, \ldots, n-1 \end{aligned}$$

Implementieren Sie diese Zerlegung. Erweitern Sie dazu die Klasse `Matrix` (siehe Homepage) um die Methode `public Matrix chol()`. Führen Sie anschließend eine Aufwandsabschätzung ihrer Implementierung in Abhängigkeit von n durch.

Aufgabe 10.2

Betrachten Sie die folgende Funktion $f : \mathbb{N}_0 \to \mathbb{N}_0$ ($\mathbb{N}_0$ sind die natürlichen Zahlen inklusive 0):

$$f(n) = \begin{cases} 0 & , n = 0 \\ 1 & , n = 1 \\ f(n-1) + f(n-2) & , n > 1 \end{cases}$$

1. Schreiben Sie eine rekursive Java-Methode, die die Funktion f implementiert.
2. Schreiben Sie eine *nicht*-rekursive Java-Methode, die die Funktion f implementiert.

[1] siehe auch „Taschenbuch der Mathematik “, Bronstein, Semendjajew, Musiol und Mühlig

Aufgabe 10.3

Betrachten Sie den folgenden Algorithmus:

```
static int f(int x, int y) {

  if (y > x) {
    return f(y, x);    // Zeile 4
  }
  int z = x % y;

  if (z == 0) {
    return y;          // Zeile 9
  } else {
    return f(y, z);    // Zeile 11
  }
}
```

Zeichnen Sie die Activation Records für den Aufruf $f(12, 21)$ zu dem Zeitpunkt, an dem die maximale Rekursionstiefe erreicht ist.