Dirk Praetorius
Christoph Überhuber

Sommersemester 07
25. April 2007

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 6

Die Aufgaben mit Stern (*) sind bis zur nächsten Übung vorzubereiten und werden dort abgeprüft. Die übrigen Aufgaben dienen nur Ihrer Übung und mir als zusätzliche Grundlage für den Prüfungsstoff in den schriftlichen Tests. Kopieren Sie bitte den Source-Code in ein Unterverzeichnis `serie06` *Ihres Home-Verzeichnisses. Überprüfen Sie bitte vor der Übung, ob Ihre Source-Codes mit dem* `gcc` *compiliert werden können. In den folgenden Aufgaben sollen im wesentlichen* ***Strukturen*** *und* ***Umgang mit dynamischem Speicher*** *geübt werden. Außer Zählschleifen, Verzweigungen und elementarer Arithmetik sind keine weiteren Elemente von C nötig.*

Aufgabe 51*. Man schreibe eine Struktur `polynomial` zur Speicherung von Polynomen, die bezüglich der Monombasis dargestellt sind, d.h. $p(x) = \sum_{j=0}^{n} a_j x^j$. Es ist also der Grad $n \in \mathbb{N}_0$ sowie der Koeffizientenvektor $a_0, \dots, a_n \in \mathbb{R}$ zu speichern. Schreiben Sie alle nötigen Funktionen, um mit dieser Struktur arbeiten zu können (`newPoly`, `delPoly`, `getPolyDegree`, `getPolyCoefficient`, `setPolyCoefficient`). Der Source-Code der Aufgaben 51–54 soll als Bibliothek `polynomial.c` (+ Header-File `polynomial.h`) im Verzeichnis `serie05` gespeichert werden.

Aufgabe 52*. Die Summe $r = p + q$ zweier Polynome p, q ist wieder ein Polynom. Man schreibe eine Funktion `addPolynomials`, die die Summe r berechnet. Zur Speicherung verwende man die Struktur aus Aufgabe 51. Zum Test schreibe man eine Funktion, die zwei Polynome einliest und deren Summe ausgibt.

Aufgabe 53*. Das Produkt $r = pq$ zweier Polynome p, q ist wieder ein Polynom. Man schreibe eine Funktion `multiplyPolynomials`, die das Produkt r berechnet. Zur Speicherung verwende man die Struktur aus Aufgabe 51. Zum Test schreibe man eine Funktion, die zwei Polynome einliest und deren Produkt ausgibt.

Aufgabe 54*. Die k-te Ableitung $p^{(k)}$ eines Polynoms p ist wieder ein Polynom. Man schreibe eine Funktion `differentiatePolynomial`, die zu gegebenem p und $k \in \mathbb{N}$ die Ableitung $p^{(k)}$ berechnet. Zur Speicherung verwende man die Struktur aus Aufgabe 51. Zum Test schreibe man eine Funktion, die p und k einliest und $p^{(k)}$ ausgibt.

Aufgabe 55. Man schreibe eine Funktion `evalPolynomial`, die den Funktionswert $p(x)$ zurückgibt. Naive Realisierung führt auf die Schachtelung zweier Schleifen und damit $\mathcal{O}(n^2)$ viele Rechenoperationen. Man überlege sich einen weniger naiven Algorithmus, der die Auswertung mittels *einer* Schleife durchführt, d.h. in $\mathcal{O}(n)$ arithmetischen Operationen. Dazu hebe man x in der Darstellung $p(x) = \sum_{k=0}^{n} a_k x^k$ geeignet heraus (sog. *Horner-Schema*).

Aufgabe 56. Man schreibe eine Struktur `vector` zur Speicherung von `double`-Vektoren der Länge n. Die Struktur enthalte neben der Dimension n den dynamischen Datenvektor. Ferner schreibe man die zugehörigen Funktionen `newVector`, `delVector`, `getVectorLength`, `getVectorEntry`, `setVectorEntry`.

Aufgabe 57. Man schreibe eine Struktur `matrix` zur Speicherung von quadratischen $n \times n$ `double` Matrizen, in der neben vollbesetzten Matrizen (Typ `0`) auch untere (Typ `'L'`) und obere (Typ `'U'`) Dreiecksmatrizen gespeichert werden können. Dabei werde die vollbesetzte Matrix im Fortran-Format spaltenweise als dynamischer Vektor der Länge $n \cdot n$ gespeichert. Dreiecksmatrizen sollen in einem Vektor der Länge $\sum_{j=1}^{n} j = n\cdot(n+1)/2$ gespeichert werden. Man schreibe die Funktionen, um mit dieser Struktur arbeiten zu können (`newMatrix`, `delMatrix`, `getMatrixDimension`, `getMatrixType`, `getMatrixEntry`, `setMatrixEntry`).

Aufgabe 58. Man schreibe eine Funktion `matrixvektor` zur Berechnung des Matrix-Vektor-Produkts $Ax \in \mathbb{R}^n$, wobei die Matrix $A \in \mathbb{R}^{n\times n}$ und der Vektor $x \in \mathbb{R}^n$ in den Datenstrukturen aus Aufgabe 56–57 gespeichert seien. Schreiben Sie die Funktion möglichst effizient, d.h. eventuelle Struktur (Dreiecksmatrix!) von A soll ausgenutzt werden.

Aufgabe 59. Nicht jede Matrix $A \in \mathbb{R}^{n\times n}$ hat eine normalisierte LU-Zerlegung $A = LU$, d.h.

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ \ell_{21} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \ell_{n1} & \dots & \ell_{n,n-1} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & u_{n-1,n} \\ 0 & \dots & 0 & u_{nn} \end{pmatrix}.$$

Wenn aber A eine normalisierte LU-Zerlegung besitzt, so gilt

$$\begin{aligned} u_{ik} &= a_{ik} - \sum_{j=1}^{i-1} \ell_{ij} u_{jk} \quad \text{für } i = 1, \dots, n, \quad k = i, \dots, n, \\ \ell_{ki} &= \frac{1}{u_{ii}} \Big(a_{ki} - \sum_{j=1}^{i-1} \ell_{kj} u_{ji} \Big) \quad \text{für } i = 1, \dots, n, \quad k = i+1, \dots, n, \\ \ell_{ii} &= 1 \quad \text{für } i = 1, \dots, n, \end{aligned}$$

wie man leicht über die Formel für die Matrix-Matrix-Multiplikation zeigen kann. Alle übrigen Einträge von $L, U \in \mathbb{R}^{n\times n}$ sind Null. Man schreibe eine Funktion `computeLU`, die die LU-Zerlegung von A berechnet und zurückgibt. Dazu überlege man, in welcher Reihenfolge man die Einträge von L und U berechnen muss, damit die angegebenen Formeln wohldefiniert sind (d.h. alles was benötigt wird, ist bereits zuvor berechnet worden). Die Matrizen sollen in der Struktur aus Aufgabe 57 gespeichert werden.

Aufgabe 60. Man schreibe eine Funktion `solve`, die die Lösung x des linearen Gleichungssystem $Ax = b$ berechnet. Für die Vektoren und die Matrix verwende man die Formate aus Aufgabe 56–57. Dabei soll die Struktur des Gleichungssystems (Dreiecksmatrix!) ausgenutzt werden. Ist A keine Dreiecksmatrix, berechne man die LU-Zerlegung und löse die Faktorisierung $LUx = b$ in zwei Schritten: (1) $Ly = b$, (2) $Ux = y$.

In der nächsten Woche wird am 02.05.2007 keine neue Übungsserie ausgegeben (Serie 7 folgt am 09.05.2007). In der Übung am 09/10.05.2007 wird stattdessen der C-Test vom 04.05.2007 besprochen!