Dirk Praetorius,
Gregor Gantner

Sommersemester 2018
12.06.2018

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 11

Aufgabe 11.1. Eine untere Dreiecksmatrix $L \in \mathbb{R}^{n\times n}$ mit

$$L = \begin{pmatrix} \ell_{11} & & & & \mathbf{0} \\ \ell_{21} & \ell_{22} & & & \\ \ell_{31} & \ell_{32} & \ell_{33} & & \\ \vdots & \vdots & \vdots & \ddots & \\ \ell_{n1} & \ell_{n2} & \ell_{n3} & \dots & \ell_{nn} \end{pmatrix}$$

hat höchstens $\frac{n(n+1)}{2} = \sum_{j=1}^{n} j$ nicht-triviale Einträge. Schreiben Sie eine Klasse `matrixL`, in der neben der Dimension $n \in \mathbb{N}$ die Koeffizienten L_{ij} in einem dynamischen Vektor der Länge $\frac{n(n+1)}{2}$ gespeichert werden. Speichern Sie L zeilenweise. Implementieren Sie die folgenden Funktionalitäten:

- Konstruktor, Copy-Konstruktor, Destruktor,
- Zuweisungssoperator,
- Zugriff auf die Koeffizienten mittels `L(i,j)` und
- die Möglichkeit eine untere Dreiecksmatrix `L` mit `cout << L` auszugeben.

Schreiben Sie auch ein main-Programm, in welchem Sie die Implementierung testen.

Aufgabe 11.2. Überladen Sie den Operator `+` für die Klasse `MatrixL` aus Aufgabe 11.1 um zwei untere Dreiecksmatrizen bei passenden Dimensionen addieren zu können. Schreiben Sie auch ein main-Programm, in welchem Sie die Implementierung testen.

Aufgabe 11.3. Überladen Sie den Operator `*` um mittels `y=L*x` das Matrix-Vektor-Produkt einer unteren Dreiecksmatrix `L` mit einem Vektor `x` berechnen zu können. `L` sei hier vom Typ `MatrixL` aus Aufgabe 11.1 und `x` ein Objekt der Klasse `Vector` aus der Vorlesung (vgl. Folie 306ff). Greifen Sie nur auf nicht-triviale Einträge der Matrix `L` zu! Schreiben Sie auch ein main-Programm, in welchem Sie die Implementierungen testen.

Aufgabe 11.4. Beweisen Sie mit der Formel des Matrix-Matrix-Produktes, dass das Produkt zweier unterer Dreiecksmatrizen eine untere Dreiecksmatrix ist. Überladen Sie den Operator `*` für die Klasse `MatrixL` aus Aufgabe 11.1 sodass Sie das Matrixprodukt für zwei untere Dreiecksmatrizen bei passenden Dimensionen berechnen können. Schreiben Sie auch ein main-Programm, in welchem Sie die Implementierungen testen.

Aufgabe 11.5. Gegeben sei eine untere Dreiecksmatrix $L \in \mathbb{R}^{n\times n}$ mit $\ell_{jj} \neq 0$ für alle $j = 1, \dots, n$. Zu gegebenem $b \in \mathbb{R}^n$ existiert dann ein eindeutiges $x \in \mathbb{R}^n$ mit $Lx = b$. Implementieren Sie die Möglichkeit, für eine untere Dreiecksmatrix $L \in \mathbb{R}^{n\times n}$ und einen Vektor $b \in \mathbb{R}^n$ das System $Lx = b$ mittels `x=L|b` zu lösen. L ist dabei vom Typ `Matrix` aus Aufgabe 11.1 und b ist dabei vom bekannten Typ `Vector` aus der Vorlesung. Schreiben Sie auch ein main-Programm, in welchem Sie die Implementierung testen.

Aufgabe 11.6. Welchen Aufwand hat das Lösen eines linearen Gleichungssystems aus Aufgabe 11.5? Schreiben Sie das Ergebnis in der $\mathcal{O}$-Notation auf und erklären Sie wie sie auf das Ergebnis gekommen sind.

Aufgabe 11.7. Adaptieren Sie gegebenenfalls den Code der Klasse `MatrixL` aus Aufgabe 11.1 so, dass Sie `new` bzw. `delete` anstatt `malloc` bzw. `free` verwenden. Was sind die Unterschiede zwischen `new` bzw. `delete` und `malloc` bzw. `free`? Was besagt die "Dreierregel"? Warum ist sie in diesem Zusammenhang wichtig?

Aufgabe 11.8. Schreiben Sie ein `Makefile` für die aktuelle Übungsserie. Dieses soll zumindest folgende Dinge umfassen:

- Erzeugen von Programmen aller von Ihnen gelösten Aufgaben.
- Das Generieren einer Bibliothek und deren Verwendung in einem Programm.