Dirk Praetorius,
Gregor Gantner

Sommersemester 2018
12.06.2018

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 11

Aufgabe 11.1. A lower triangular matrix $L \in \mathbb{R}^{n\times n}$ with

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

has at most $\frac{n(n+1)}{2} = \sum_{j=1}^{n} j$ nontrivial coefficients. Write a class `matrixL` to save the coefficients L_{ij} in a dynamical vector with length $\frac{n(n+1)}{2}$ together with the dimension $n \in \mathbb{N}$. Save the matrix L row-wise. Implement the following features:

- constructor, copy-constructor, destructor,
- assignment operator,
- access to the coefficients via `L(i,j)` and
- the possibility to print a lower triangular matrix `L` on screen via `cout << L`.

Moreover, write a main-program to test your implementation.

Aufgabe 11.2. Overload the operator `+` for the class `MatrixL` from Exercise 11.1 to be able to add to lower triangular matrices with matching dimensions. Moreover, write a main-programm to test your implementation.

Aufgabe 11.3. Overload the operator `*` to compute the matrix-vector-product `y=L*x` of a lower triangular matrix `L` and a vector `x`. Here, let `L` be of the type `MatrixL` from exercise 11.1 and `x` an object of the class `Vector` from the lecture (cf. Slide 306ff). Access non-trivial entries of the matrix `L` only! Moreover, write a main program in order to test your implementation accurately.

Aufgabe 11.4. Use the formula for the matrix-matrix product to show that the product of two lower triangular matrices is a lower triangular matrix. Then, overload the operator `*` for the class `MatrixL` from Exercise 11.1 to be able to perform the matrix-matrix product for two lower triangular matrices with matching dimensions. Moreover, write a main-program to test your implementation.

Aufgabe 11.5. Let $L \in \mathbb{R}^{n\times n}$ be a lower triangular matrix such that $\ell_{jj} \neq 0$ for all $1 \le j \le n$. Given $b \in \mathbb{R}^n$, there exists a unique $x \in \mathbb{R}^n$ such that $Lx = b$. Implement also the feature to solve the system $Lx = b$ for a lower triangular matrix $L \in \mathbb{R}^{n\times n}$ and a vector $b \in \mathbb{R}^n$ by using the command `x=L|b`. L has the type `MatrixL` from Exercise 11.1 and b has the well-known type `Vector` from the lecture. Moreover, write a main-program to test your implementation.

Aufgabe 11.6. What is the computational cost to solve a linear equation system like in exercise 11.5? Write down your results in the $\mathcal{O}$-notation.

Aufgabe 11.7. Adapt the Code from the class `MatrixL` from Exercise 11.1, so that `new` resp. `delete` is used instead of `malloc` resp. `free` (if you have not already implemented it that way). What are the differences between `new` resp. `delete` and `malloc` resp. `free`? What is the "Rule of three" saying? Why is this rule important in that context?

Aufgabe 11.8. Write a `Makefile` for the exercises of this sheet. It should contain:

- The compilation of all solved exercises.
- The generation of a library and an example of its usage.