Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825
Institut für Angewandte und Numerische Mathematik

Programmieren: Einstieg in die Informatik mit Java

WS 2006/2007

Dr. G. Bohlender
Dipl.–Math. techn. M. Richter

29.01.2007

Aufgabenblatt 12

Aufgabe 29: *Rechnen mit dünnbesetzten Matrizen (Klausuraufgabe WS 2005/2006)*

Schreiben Sie in Java eine öffentliche Klasse namens `MatrixDB`, in der die nachstehenden Operationen für *dünnbesetzte Matrizen* umgesetzt werden.

Eine Matrix $\boldsymbol{A} \in \mathbb{R}^{n,n}$, $n \in \mathbb{N}$, heißt *dünnbesetzt*, falls die Anzahl der von Null verschiedenen Elemente viel kleiner als n^2 ist.
Sei nun $m \in \mathbb{N}_0$ die Anzahl der von Null verschiedenen Elemente der Matrix $\boldsymbol{A}$, sei weiterhin $\alpha_k \in \mathbb{R}_{\neq 0}$ für $k \in \{0, 1, \ldots, m-1\}$ das $(k+1)$-te von Null verschiedene Element der Matrix $\boldsymbol{A}$ und sei (i_k, j_k) mit $i_k, j_k \in \{0, 1, \ldots, n-1\}$ die Position des Elementes α_k in der Matrix $\boldsymbol{A}$. Dann läßt sich die Matrix $\boldsymbol{A}$ alternativ zur Anordnung als quadratisches Zahlenschema

$$\boldsymbol{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \ldots & a_{0,n-1} \\ a_{1,0} & a_{1,1} & \ldots & a_{1,n-1} \\ \vdots & \vdots & & \vdots \\ a_{n-1,0} & a_{n-1,1} & \ldots & a_{n-1,n-1} \end{bmatrix} = \left[a_{p,q}\right]_{p,q=0,1,\ldots,n-1} \tag{1}$$

über die folgenden drei Vektoren $\boldsymbol{i}, \boldsymbol{j} \in \{0, 1, \ldots, n-1\}^m$ und $\boldsymbol{\alpha} \in \mathbb{R}^m_{\neq 0}$ darstellen:

$$\boldsymbol{i} = [i_0; i_1; \ldots; i_{m-1}], \tag{2}$$
$$\boldsymbol{j} = [j_0; j_1; \ldots; j_{m-1}], \tag{3}$$
$$\boldsymbol{\alpha} = [\alpha_0; \alpha_1; \ldots; \alpha_{m-1}]. \tag{4}$$

Es gilt dabei: $\alpha_k = a_{i_k, j_k}$ für $k \in \{0, 1, \ldots, m-1\}$.

Die Matrix

$$\boldsymbol{A} := \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \in \mathbb{R}^{4,4} \tag{5}$$

kann somit durch die Vektoren

$$\boldsymbol{i} = [0; 0; 1; 1; 1; 2; 2; 2; 3; 3], \tag{6}$$
$$\boldsymbol{j} = [0; 1; 0; 1; 2; 1; 2; 3; 2; 3], \tag{7}$$
$$\boldsymbol{\alpha} = [2; -1; -1; 2; -1; -1; 2; -1; -1; 2] \tag{8}$$

dargestellt werden.

Man definiert die Multiplikation einer Matrix mit einem Vektor wie gewohnt als:

$$* : \quad \mathbb{R}^{n,n} \times \mathbb{R}^n \longrightarrow \mathbb{R}^n, \quad (\boldsymbol{A}, \boldsymbol{v}) \mapsto \boldsymbol{A} * \boldsymbol{v} := \left[\sum_{j=0}^{n-1} a_{i,j} v_j\right]_{i=0,1,\ldots,n-1}. \tag{9}$$

Ist die Matrix $\boldsymbol{A}$ durch die drei Vektoren $\boldsymbol{i}$, $\boldsymbol{j}$ und $\boldsymbol{\alpha}$ gegeben, dann kann das in Gleichung (9) definierte Produkt $\boldsymbol{w} := \boldsymbol{A} * \boldsymbol{v}$ mit Hilfe des nachfolgend angegebenen Verfahrens berechnet werden:

Initialisierung: $w_l = 0$ für $l \in \{0, 1, \ldots, n-1\}$, (10)

Iteration für $k = 0, 1, \ldots, m-1$:

$$s = w_{i_k}, \tag{11}$$
$$w_{i_k} = s + \alpha_k v_{j_k}. \tag{12}$$

Verwenden Sie bei der Umsetzung die folgenden Elemente:

- Eine private Instanzvariable namens `n` vom Typ `int` zur Darstellung der Dimension $n \in \mathbb{N}$ einer dünnbesetzten Matrix $\boldsymbol{A} \in \mathbb{R}^{n,n}$.
- Eine private Instanzvariable namens `m` vom Typ `int` zur Darstellung der Anzahl der von Null verschiedenen Elemente der Matrix $\boldsymbol{A}$.
- Zwei private Instanzvariablen namens `i` und `j` vom Typ `int[]` zur Darstellung der Vektoren $\boldsymbol{i}$ und $\boldsymbol{j}$
- Eine private Instanzvariable namens `alpha` vom Typ `double[]` zur Darstellung des Vektors $\boldsymbol{\alpha}$.
- Einen öffentlicher Konstruktor mit zwei Parametern vom Typ `int` für die Dimension $n \in \mathbb{N}$ der anzulegenden Matrix und die Anzahl $m \in \mathbb{N}_0$ der von Null verschiedenen Elemente in der Matrix, in dem die Instanzvariablen `n` und `m` mit den entsprechenden Werten der übergebenen Parameter initialisiert werden und dementsprechend der Speicher für die Komponenten der Vektoren $\boldsymbol{i}$, $\boldsymbol{j}$ und $\boldsymbol{\alpha}$ beschafft und den Instanzvariablen `i`, `j` und `alpha` zugewiesen wird.
- Eine Klassenmethode `lies` mit einem Parameter vom Typ `int` zur Darstellung der Dimension $n \in \mathbb{N}$ des einzulesenden Vektors $\boldsymbol{v} \in \mathbb{R}^n$, in der die Komponenten des übergebenen Vektors von der Konsole eingelesen und in Form einer Variablen vom Typ `double[]` an die aufrufende Methode zurückgeliefert werden.
- Eine Instanzmethode `lies` ohne Parameter, in der in einer `for`–Schleife über k alle von Null verschiedenen Elemente α_k einer dünnbesetzten Matrix zusammen mit ihrer Position (i_k, j_k) von der Konsole eingelesen werden.
  Speichern Sie dabei für $k \in \{0, 1 \ldots, m-1\}$ das $(k+1)$-te von Null verschiedene Element der einzulesenden Matrix in der k-ten Komponente der Instanzvariablen `alpha` ab. Verfahren Sie dabei analog mit den Instanzvariablen `i` und `j`.

- Eine Instanzmethode `zeige` ohne Parameter, in der in einer `for`–Schleife alle von Null verschiedenen Elemente einer dünnbesetzten Matrix zusammen mit ihrer Position in der Matrix mit Hilfe der Methode `System.out.println` zeilenweise auf der Konsole ausgegeben werden.

- Eine Klassenmethode `zeige` mit einem Parameter vom Typ `double[]` zur Darstellung eines Vektors, in der die Elemente des übergebenen Vektors zeilenweise auf der Konsole ausgegeben werden.

- Eine Instanzmethode `mult` mit einem Parameter vom Typ `double[]` zur Darstellung eines Vektors, in der das Produkt einer dünnbesetzten Matrix mit dem übergebenen Vektor gemäß dem in den Gleichungen (10) bis (12) angegebenen Verfahren berechnet und das Ergebnis in Form einer Variablen vom Typ `double[]` an die aufrufende Methode zurückgeliefert wird.

- Ein Hauptprogramm, in dem zu Beginn die Dimension $n \in \mathbb{N}$ und die Anzahl $m \in \mathbb{N}_0$ von Null verschiedenen Elemente einer dünnbesetzten Matrix $\boldsymbol{A} \in \mathbb{R}^{n,n}$ von der Konsole eingelesen werden, die Matrix $\boldsymbol{A}$ mittels des Konstruktors erzeugt und unter Zuhilfenahme der Methoden `lies` von der Konsole eingelesen wird, anschließend die Matrix mit Hilfe der Methode `zeige` wieder auf der Konsole ausgegeben wird, danach ein Vektor mittels der Methode `lies` von der Konsole eingelesen wird, und zum Schluß das Produkt der Matrix $\boldsymbol{A}$ mit dem eingelesenen Vektor unter Verwendung der Methoden `mult` und `zeige` berechnet und auf der Konsole ausgegeben wird.