Technische Universität Braunschweig

Institut für
Betriebssysteme und Rechnerverbund

Prof. Dr. H. Langendörfer · Prof. Dr. M. Zitterbart

Übungen zu Algorithmen und Datenstrukturen II

Blatt 2

3. Aufgabe: Emulation mehrdimensionaler Arrays

In einem Rechner sind intern alle Daten immer eindimensional (linear) abgelegt. Eine Programmiersprache (z.B. Java), die die Möglichkeit von mehrdimensionalen Arrays bietet, muß daher ein mehrdimensionales Array durch ein eindimensionales Array nachbilden. Dazu wird der n-dimensionale Index $(i_1, \ldots, i_n)$, der ein Element im Array bestimmt, eineindeutig auf einen Index des eindimensionalen Arrays abgebildet.

1. Geben Sie eine Funktion $f(i_1, i_2)$ für die Indextransformation eines zweidimensionalen Arrays an. Der Index der Dimension j $(j = \{1, 2\})$ läuft dabei von $0, \ldots, m_j - 1$.
2. Verallgemeinern Sie die Formel für n-dimensionale Arrays.
3. Ein Sonderfall eines 2-dimensionalen Arrays ist ein Feld in Dreiecksform. Dieses wird häufig in der Numerik zur Speicherung dreiecksförmiger Matrizen benötigt. Das dreiecksförmige Array mit den Elementen a_{i_1,i_2} hat dabei folgenden Aufbau:

$$\begin{pmatrix} a_{0,0} & & & \\ a_{1,0} & a_{1,1} & & \\ \cdots & & & \\ a_{n-1,0} & a_{n-1,1} & \ldots & a_{n-1,n-1} \end{pmatrix}$$

Geben Sie eine Formel für die Speicherung eines dreiecksförmigen Arrays an.

4. Aufgabe: Stacks

Erstellen Sie ein Java-Programm zur iterativen Berechnung der folgenden verallgemeinerten Binomialkoeffizienten.

$$\begin{array}{ccccccccc} & & & & 1 & & & & \\ & & & 1 & & 2 & & & \\ & & 1 & & 3 & & 4 & & \\ & 1 & & 4 & & 7 & & 8 & \\ 1 & & 5 & & 11 & & 15 & & 16 \end{array}$$

Das allgemeine Bildungsgesetz lautet:

$$\binom{d}{0} = 1, \binom{d}{d} = 2^d, \binom{d}{h} = \binom{d-1}{h} + \binom{d-1}{h-1}$$

Verwenden Sie zur Realisierung die Schnittstelle des in der Vorlesung definierten abstrakten Datentyps Stack.

5. Aufgabe: Stacks (zum Zweiten)

Ein vollständig geklammerter, arithmetischer Ausdruck ist ein Term der Form $(\mu + \nu), (\mu - \nu), (\mu * \nu)$ oder (μ / ν), wobei die Variablen μ und ν ebenfalls wiederum vollständig geklammerte, arithmetische Ausdrücke sein können. Beispiel für einen solchen Ausdruck ist der Term $(\mu - ((\nu + \xi) + \rho))$.

(a) Geben Sie ein Programm zur Auswertung von vollständig geklammerten, arithmetischen Ausdrücken unter Verwendung eines Stacks an. Für die Bearbeitung und Auswertung der Ausdrücke darf Pseudocode verwendet werden, insbesondere die Stack-Operationen aber sind wie in der Vorlesung zu verwenden.

Die obige Notation vollständig geklammerter, arithmetischer Ausdrücke heißt *Infixnotation*. Es existiert eine weitere Notation, die sogenannte *Postfixnotation*. Aus $(\mu + \nu)$ in Infixnotation wird $\mu\nu+$ in Postfixnotation, aus $(\mu * \nu)$ wird $\mu\nu*$ etc. Der obige Beispielterm $(\mu - ((\nu + \xi) + \rho))$ lautet in Postfixnotation $\mu\nu\xi + \rho + -$.

(b) Entwerfen Sie ein Programm, welches einen vollständig geklammerten, arithmetischen Ausdruck in Infixnotation in sein Äquivalent in Postfixnotation unter Verwendung des Datentyps Stack umwandelt. Weiterhin gilt das unter (a) gesagte.