Prof. Dr. A. Goerdt
Dr. L. Falke

TU Chemnitz
Wintersemester 2017/2018
17.01.2018

Theoretische Informatik I

12. Übung

Geben Sie die Lösung der Aufgabe *2* bitte bis zum *22.01.2018 9:15 Uhr* ab. (Briefkasten vorm Raum 1/266 oder per eMail an `falu@informatik.tu-chemnitz.de`, *Betreff:* `TI1 Hausaufgaben`)

1. Aufgabe: Gegeben sind die beiden Zeichenfolgen A und B.

$$A = \text{cdbaeg}\ , \quad B = \text{abdgae}$$

Berechnen Sie die *längste gemeinsame Teilfolge* von A und B. Verwenden Sie dazu das Verfahren zur dynamischen Programmierung aus der Vorlesung.

2. Aufgabe: Analog zur *längsten gemeinsamen Teilfolge* läßt sich auch eine *kürzeste gemeinsame Oberfolge* definieren: Gegeben sind zwei Zeichenfolgen A und B. Wir suchen eine *kürzeste* Zeichenfolge C, so dass sowohl A als auch B *Teilfolgen* von C sind.

Wir betrachten zum Beispiel die Zeichenfolgen

$$A = \text{abec} \quad \text{und} \quad B = \text{dbc}\ .$$

Dann sind

$$C_1 = \text{adbec} \quad \text{aber auch} \quad C_2 = \text{dabec}$$

kürzeste gemeinsame Oberfolgen von A und B.

(a) Geben sie einen *rekursiven Ansatz* zur Lösung dieses Problems an. Gehen Sie analog zu den Überlegungen zum Fall der *längsten gemeinsamen Teilfolge* in der Vorlesung vor.

(b) Formulieren Sie einen Algorithmus, der mit Hilfe dynamischer Programmierung die Länge der kürzesten gemeinsamen Oberfolge in Zeit $O(|A| \cdot |B|)$ bestimmt.

(c) Geben Sie die Tabelle, so wie sie von ihrem Algorithmus ausgefüllt wird, an.

3. Aufgabe: Die Multiplikation von Matrizen ist assoziativ, d.h. die Rechnung $(A \cdot B) \cdot C$ liefert das gleiche Ergebnis wie die Rechnung $A \cdot (B \cdot C)$. Allerdings sind je nach Klammerung unterschiedlich viele skalare Multiplikationen notwendig.

Ein Beispiel. Die 3×5-Matrix A, die 5×3-Matrix B und die 3×3-Matrix C.

$$A = \begin{pmatrix} \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \end{pmatrix} \quad B = \begin{pmatrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{pmatrix} \quad C = \begin{pmatrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{pmatrix}$$

Die Rechnung $(A \cdot B) \cdot C$ benötigt $3^2 \cdot 5 + 3^2 \cdot 3 = 72$ Multiplikationen, während $A \cdot (B \cdot C)$ $(5 \cdot 3) \cdot 3 + 3^2 \cdot 5 = 90$ Multiplikationen benötigt.

Sind nun n Matrizen $M_1, M_2, \ldots, M_n$ gegeben, interessiert man sich für die Klammerung des Produktes $M_1 \cdot M_2 \cdot \ldots \cdot M_n$, bei der die wenigsten Multiplikationen ausgeführt werden müssen.

(a) Geben Sie zunächst einen rekursiven Lösungsansatz für das Problem an.

Wichtig: Wie berechnet sich die Anzahl der notwendigen Multiplikationen für die betrachteten Teilstücke?

(b) Entwerfen Sie daraus einen Algorithmus, der dieses Problem in einer Laufzeit von $O(n^3)$ löst. Verwenden Sie dazu den Ansatz der *dynamischen Programmierung*.

(c) Welche bedeutung haben Ihre Tabellenenträge und in welcher Reihenfolge müssen diese ausgefüllt werden?

Geben Sie am Ende auch eine optimale Klammerung aus.

Die Eingabe ist ein Feld $D[0..n]$ mit den Dimensionen der Matrizen. Dabei steht $D[i-1]$ für die Anzahl der Zeilen vom M_i und $D[i]$ für die Anzahl der Spalten von M_i. Im Beispiel wäre D also $[3, 5, 3, 3]$.

Hinweis: Die *Matrixkettenmultiplikation* ist dem Problem des *optimalen statischen binären Suchbaumes* aus der Vorlesung sehr ähnlich.