

41. Matrizenmultiplikation, 10 Punkte

Bei der Matrizenmultiplikation kommt es stark auf die Reihenfolge der ausgeführten Teilmultiplikationen an.

Teilprobleme

Das Problem lässt sich in zwei Teilfragen näher betrachten:

Ziel 1: finden der minimalen Anzahl von Operationen, um Matrizen zu multiplizieren.

Ziel 2: finden der optimalen Reihenfolge von Matrixmultiplikationen, um die Anzahl an Operationen zu minimieren.

Findet man eine Antwort zum ersten Ziel hat man bereits eine Lösung für das zweite Ziel gefunden. Es ist nur wichtig die Rechenschritte die man bei der Lösung von 1 gemacht hat entsprechend zu notieren. Dafür hilft man sich mit einer Matrix $Z(k \times k)$.

Rekursion

Die Felder der Matrix Z werden mit der Anzahl an Operationen gefüllt, die man benötigt, um die Matrix A_i mit der Matrix A_k zu multiplizieren. Folgende Rekursionsgleichung wird verwendet:

$$Z[i, j] = \begin{cases} 0, & \text{wenn } i = j \\ \min_{l=i \rightarrow k-1} \{Z(i, l) + Z(l+1, k) + m_i * n_l * p_k\} \end{cases}$$

Die Rekursionsgleichung soll auf bereits berechnete Ergebnisse zurückgreifen können. Daher sollte die Berechnung an der Diagonalen beginnen (weil $Z[i, j] = 0 \mid i = j$) und sich anschließend zur Optimallösung an Position $Z[1, k]$ oder $Z[k, 1]$ "vorarbeiten".

Zur Veranschaulichung die Matrix mit $k = 4$: $Z =$

i,j	1	2	3	4
1	0	$Z[1, 2]$	$Z[1, 3]$	$Z[1, 4]$
2		0	$Z[2, 3]$	$Z[2, 4]$
3			0	$Z[3, 4]$
4				0

Algorithmus

1. erzeuge Matrix $Z(k \times k)$ mit Nullen gefüllt
2. berechne die Werte der Matrix diagonal, um zur Ecke rechts oben oder links unten zu gelangen
3. für jedes Feld berechne $Z[i, j] = \begin{cases} 0, & \text{wenn } i = j \\ \min_{l=i \rightarrow k-1} \{Z(i, l) + Z(l+1, k) + m_i * n_l * p_k\} \end{cases}$ rekursiv.
Notiere den Wert von l , für den das Minimum berechnet wurde.

Laufzeit

Um die Matrix Z zu berechnen sind $\Theta(\frac{1}{2}n^2)$ Schritte notwendig. In jedem Schritt wird die Rekursionsgleichung $Z[i, j]$ berechnet und diese braucht $\Theta(n)$ Zeit. Die Gesamtlaufzeit ist also vereinfacht (ohne $c = \frac{1}{2}$) $\Theta(n * n^2) = \Theta(n^3)$ Zeit.

43. Optimaler binärer Suchbaum, 10 Punkte

Algorithmus aus VL

Anmerkung: In der VL wurden nur die Gleichungen gegeben. Im offiziellen Skript ist bis zum 1.12., 7 Uhr (Zeit meiner letzten Einsicht) kein Algorithmus angegeben!!! Der Algorithmus wurde auch nicht im Tutorium besprochen, wie nie etwas vorbesprochen wird! (An dieser Stelle hätte es durchaus mal Sinn gemacht!)

Teilprobleme

$f(i, j) \quad 0 \leq i \leq j \leq n + 1$ optimaler Suchbaum für Schlüssel x_{i+1}, \dots, x_{j-1} und Häufigkeiten $p_{i+1}, \dots, p_j - 1$

Rekursion

$$f(i, j) = \min\{f(i, k) + f(k + 1, j)\} + q_i + q_{i+1} + \dots + q_{j-1} + p_{i+1} + p_{i+2} + \dots + p_{j-1} - p_k \quad | i + 1 \leq k \leq j - 1, 0 \leq i, j \leq n + 1, j \geq i + 1$$

Anfangswerte

$$f(i, i + 1) = 0, i = 0, \dots, n$$

Beispiel: $f(2, 4) = f(2, 3) + f(3, 4) + q_2 + q_3 + p_3 - p_3$

Gesamtlösung

$$f(0, n + 1)$$

Berechnung der Lösung

Gegeben

$$\begin{aligned}(S_1, S_2, S_3) &= (10, 12, 17) \\ (p_1, p_2, p_3) &= (8, 1, 4) \\ (q_0, q_1, q_2, q_3) &= (1, 2, 10, 4) \\ n &= 3, S_0 = -\infty, S_4 = \infty\end{aligned}$$

$$f(i, j) = \min\{f(i, k) + f(k + 1, j)\} + q_i + q_{i+1} + \dots + q_{j-1} + p_{i+1} + p_{i+2} + \dots + p_{j-1} - p_k \quad | i + 1 \leq k \leq j - 1, 0 \leq i, j \leq n + 1, j \geq i + 1$$

$$\begin{aligned}f(0, 4) &= \min\{f(0, 2) + f(3, 4)\} + q_0 + q_1 + q_1 + q_3 + p_1 + p_2 + p_3 - p_2 \\ f(0, 2) &= \min\{f(0, 1) + f(2, 2)\} + q_0 + q_1 + p_1 - p_1 \\ f(3, 4) &= \min\{f(3, 4) + f(4, 4)\} + q_3 + p_4 - p_4 \\ f(0, 1) &= \min\{f(0, 1) + f(2, 1)\} + q_0 + p_1 - p_1 \\ f(2, 2) &= \min\{f(2, 2) + f(3, 2)\} + q_2 + p_3 + p_1 - p_3 \\ f(4, 4) &= \min\{f(4, 4) + f(5, 4)\} + q_4 + p_4 - p_4\end{aligned}$$