

---

## Algorithmen für die Speicherhierarchie

---

*Abgabetermin: 19.11.2007 vor der Übung*

### Algorithmen/obere Schranken

#### Aufgabe 1 (4 Punkte)

Betrachten Sie das Problem, zwei allgemeine vollbesetzte Matrizen zu multiplizieren. Die Eingabe-Matrizen haben die Größe  $N_1 \times N_2$  und  $N_2 \times N_3$ . Das Ergebnis ist also eine Matrix der Größe  $N_1 \times N_3$ .

Verallgemeinern Sie den in der Vorlesung vorgestellten Algorithmus für quadratische Matrizen auf den allgemeinen (nicht-quadratischen) Fall. Geben Sie eine passende obere Schranke an! (Und begründen sie diese.)

#### Aufgabe 2 (6 Punkte)

Gegeben sei folgendes Problem: Eingabe sind zwei Listen von Paaren  $X = (x_i, p_i)$  und  $Y = (y_i, q_i)$ . O.B.d.A. sei dabei  $|X| \leq |Y|$ . Zu berechnen ist der Wert  $z = \sum_{(i,j): |p_i - q_j| < \epsilon} x_i y_j$ . Es werden also alle Produkte aufsummiert, bei denen das jeweilige  $p$  und  $q$  um weniger als ein gegebenes  $\epsilon$  voneinander abweichen.

Beschreiben Sie hierfür einen (möglichst effizienten) Algorithmus und bestimmen Sie dessen I/O-Komplexität.

(Hinweis: Es gibt einen Algorithmus der mit  $O\left(\frac{|Y|}{B} \log_{M/B} \frac{|X|}{B} + \frac{k_\epsilon}{BM}\right)$  I/O-Operationen auskommt. Wobei  $k_\epsilon = |\{(i, j) : |p_i - q_j| < \epsilon\}|$  ist.)