Algorithmenentwurf HA 3

Lukas Brandt: 7011823, Clemens Damke: 7011488, Lukas Giesel: 7011495
28. April 2016

Aufgabe 5

a. Beweis

$$
\begin{aligned}
P &= (A_{11} + A_{22}) \cdot (B_{11} + B_{22}) && = A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22} \\
Q &= (A_{21} + A_{22}) \cdot B_{11} && = A_{21} \cdot B_{11} + A_{22} \cdot B_{11} \\
R &= A_{11} \cdot (B_{12} - B_{22}) && = A_{11} \cdot B_{12} - A_{11} \cdot B_{22} \\
S &= A_{22} \cdot (B_{21} - B_{11}) && = A_{22} \cdot B_{21} - A_{22} \cdot B_{11} \\
T &= (A_{11} + A_{12}) \cdot B_{22} && = A_{11} \cdot B_{22} + A_{12} \cdot B_{22} \\
U &= (A_{21} - A_{11}) \cdot (B_{11} + B_{12}) && = A_{21} \cdot B_{11} + A_{21} \cdot B_{12} - A_{11} \cdot B_{11} - A_{11} \cdot B_{12} \\
V &= (A_{12} - A_{22}) \cdot (B_{21} + B_{22}) && = A_{12} \cdot B_{21} + A_{12} \cdot B_{22} - A_{22} \cdot B_{21} - A_{22} \cdot B_{22}
\end{aligned}
$$

$$
\begin{aligned}
C_{11} &= P + S - T + V && = A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22} \\
&&& + A_{22} \cdot B_{21} - A_{22} \cdot B_{11} - A_{11} \cdot B_{22} - A_{12} \cdot B_{22} \\
&&& + A_{12} \cdot B_{21} + A_{12} \cdot B_{22} - A_{22} \cdot B_{21} - A_{22} \cdot B_{22} \\
&&& = A_{11} \cdot B_{11} + A_{12} \cdot B_{21} \checkmark \\
C_{12} &= R + T && = A_{11} \cdot B_{12} - A_{11} \cdot B_{22} + A_{11} \cdot B_{22} + A_{12} \cdot B_{22} \\
&&& = A_{11} \cdot B_{12} + A_{12} \cdot B_{22} \checkmark \\
C_{21} &= Q + S && = A_{21} \cdot B_{11} + A_{22} \cdot B_{11} + A_{22} \cdot B_{21} - A_{22} \cdot B_{11} \\
&&& = A_{21} \cdot B_{11} + A_{22} \cdot B_{21} \checkmark \\
C_{22} &= P + R - Q + U && = A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22} \\
&&& + A_{11} \cdot B_{12} - A_{11} \cdot B_{22} - A_{21} \cdot B_{11} - A_{22} \cdot B_{11} \\
&&& + A_{21} \cdot B_{11} + A_{21} \cdot B_{12} - A_{11} \cdot B_{11} - A_{11} \cdot B_{12} \\
&&& = A_{21} \cdot B_{12} + A_{22} \cdot B_{22} \checkmark
\end{aligned}
$$

C_{11}, C_{12}, C_{21}, C_{22} sind äquivalent zur Definition in *Aufgabe 3* und ergeben zusammengesetzt somit C. ■

Die Anzahl der benötigten Multiplikationen beträgt 7, da in jeder der sieben Hilfsvariablen P, Q, R, S, T, U, V jeweils eine Multiplkation stattfindet und sonst nirgends.

Die Anzahl der benötigten Additionen beträgt $10 + 8 = 18$, da zur Berechnung der Hilfsvariablen jeweils 2, 1, 1, 1, 1, 2, 2 ($\sum = 10$) und zur Berechnung von C_{ij} jeweils 3, 1, 1, 3 ($\sum = 8$) Additionen benötigt werden.

b. Algorithmus

Algorithm 1 Matrizenmultiplikation

Require: $\exists k \in \mathbb{N}_0, n := 2^k : A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \wedge B = \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nn} \end{bmatrix}$

function M-MULT(A, B)
  if $n = 1$ **then**
    return $a_{11} \cdot b_{11}$ ▷ $\mathcal{O}(1)$

  $A_{11}, A_{12}, A_{21}, A_{22} \leftarrow (\frac{n}{2} \times \frac{n}{2})$-Submatrizen von A ▷ $\mathcal{O}(n^2)$
  $B_{11}, B_{12}, B_{21}, B_{22} \leftarrow (\frac{n}{2} \times \frac{n}{2})$-Submatrizen von B ▷ $\mathcal{O}(n^2)$

  $P, Q, R, S, T, U, V \leftarrow$ definiert wie in Aufgabenstellung, wobei '·' durch M-MULT und '+' durch M-ADD ersetzt wird. $\implies 7 \times$ M-MULT $= \mathcal{O}(7 \cdot T(\frac{n}{2})), 10 \times$ M-ADD $= \mathcal{O}(n^2)$

  $C_{11}, C_{12}, C_{21}, C_{22} \leftarrow$ definiert wie in Aufgabenstellung, wobei '+' durch M-ADD ersetzt wird. $\implies 8 \times$ M-ADD $= \mathcal{O}(n^2)$

  return $\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$ ▷ $\mathcal{O}(n^2)$

function M-ADD(A, B)
  for all $i, j \in \{1, \ldots, n\}$ **do**
    $c_{ij} \leftarrow a_{ij} + b_{ij}$

  return $\begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nn} \end{bmatrix}$

Laufzeit:

$$T(n) = 7c \cdot T(\frac{n}{2}) + c \cdot n^2$$

$$\underset{\text{Mastertheorem } a>b}{\implies} T(n) \leq \frac{7}{7-2} c \cdot n^{\log_2 7} = \frac{7}{5} c \cdot n^{\log_2 7} = \mathcal{O}(n^{\log_2 7}) = \mathcal{O}(n^{2.8074})$$

Aufgabe 6

a. Beweis

z.z. $A \Leftrightarrow B$ mit A, B linke bzw. rechte Aussage aus Aufgabenstellung.

$$\Gamma(v, G) := \text{Nachbarn des Knotens } v \text{ im Graphen } G$$
$$pos(a) := i \text{ sodass } a = v_i$$

- $A \Rightarrow B$:

  Finde unter den ersten d_0 Knoten (gemäß einer Sortierung der Knoten nach ihren Graden) die, die nicht Nachbarn von v_0 sind ($= Y$) und welche Knoten stattdessen Nachbarn von v_0 sind ($= X$):

  $$X := \{k \in \Gamma(v_0, G) \mid pos(k) > d_0\} = \{x_1, \ldots, x_{|X|}\}$$
  $$Y := \{k \notin \Gamma(v_0, G) \mid pos(k) \leq d_0\} = \{y_1, \ldots, y_{|Y|}\}$$
  $$\implies |X| = |Y| =: z$$

  $$\sigma(y_i) := v \text{ sodass } \{y_i, v\} \in E \wedge v \neq x_i$$

  σ ist für alle y_i definiert, da:

  $\forall y_i \in Y : deg(y_i) = d_{pos(y_i)} \geq d_{pos(x_i)} = deg(x_i) \geq 1$, da x_i Nachbar von v_0 ist.

  $$\implies \begin{cases} deg(y_i) = 1 \Rightarrow deg(x_i) = 1 \Rightarrow \text{Nachbar von } y_i \text{ ist nicht } x_i, \text{ da } \Gamma(x_i, G) = \{v_0\}. \\ deg(y_i) > 1 \Rightarrow \exists \text{ Nachbar von } y_i, \text{ der nicht } x_i \text{ ist.} \end{cases}$$

  Entferne nun aus E alle Nachbarn von v_0. Der resultierende Graph entspricht dem induzierten Teilgraphen von G ohne v_0:

  $$E_0 := E \setminus \{\{v_0, v\} \mid v \in V\}$$

  E_0 erfüllt B nicht notwendigerweise, da $\forall x \in X$ der Grad von Knoten 'hinter' den ersten d_0 verringert wurde und dafür der Grad von Knoten 'vor' den ersten d_0 ($\in Y$) unverändert blieb. Um dies zu korrigieren wird von jedem Knoten aus Y eine Kante entfernt und stattdessen einem Knoten in X hinzugefügt:

  $$E' := E_0 \setminus \{\{y_i, \sigma(y_i)\} \mid i \in \{1, \ldots, z\}\} \cup \{\{x_i, \sigma(y_i)\} \mid i \in \{1, \ldots, z\}\}$$
  $$G' := (\{v_1, \ldots, v_{n-1}\}, E')$$
  $$\implies G' \text{ erfüllt } B \text{ gemäß Konstruktion. } \checkmark$$

▶ $A \Leftarrow B$:

$$E := E' \cup \{\{v_0, v_i\} \mid i \in \{1, \ldots, d_0\}\}$$
$$G := (\{v_0, \ldots, v_{n-1}\}, E)$$
$\implies$ G erfüllt A gemäß Konstruktion, da der Grad der ersten d_0 Knoten um 1 erhöht wurde, v_0 genau d_0 Kanten hat und somit $\forall v_i : deg(v_i) = d_i$ gilt. ✓

■

b. Algorithmus

Algorithm 2 Graphexistenz

Require: $d_0 \geq \cdots \geq d_n, d_i \in \mathbb{N}_0$

1: **function** GraphExists($d_0, \ldots, d_n$)
2:     **if** $d_0 < 0$ **then return** false
3:     **if** $d_0 = 0$ **then return** true

4:     **for** $i \leftarrow 0$ to $n - 1$ **do**
5:         **for** $j \leftarrow i + 1$ to $i + d_i$ **do**
6:             **if** $d_j = 0$ **then return** false
7:             $d_j \leftarrow d_j - 1$

8:         **if** $i + d_i < n \land d_{i+d_i} < d_{i+d_i+1}$ **then return** false

9:     **return** true

c. Laufzeit

Die Zeilen 2, 3, 9 benötigen konstante Zeit. Die Zeilen 6, 7 werden $\mathcal{O}(n^2)$ mal ausgeführt. Zeile 8 $\mathcal{O}(n)$ mal.

$\implies T(n) = \mathcal{O}(n^2)$