Algorithmen und Datenstrukturen

Vincent Dahmen 6689845 Roberto Seidel 6537468
Rafael Heid 6704828

8. November 2015

3.1

ALGO1 Für den ersten Algorithmus gilt:
Die äußere Schleife wird n mal betreten.
Die inneren beiden Schleifen werden jeweils n mal betreten.
Daraus folgt eine gesamte Laufzeit von $(n * (n + n))$ was in $O(n^2)$ liegt.

ALGO2 Für den zweiten Algorithmus gilt:
Die äußere Schleife wird n mal betreten (Grenze bei 2n, aber Schritt bei n+=2).
Die innere Schleife wird n mal ausgeführt (Schleife wird $2 \cdot n$ mal betreten, geht aber nur bis $\frac{n}{2}$)
Daraus folgt eine gesamte Laufzeit von $(n \cdot (2 \cdot \frac{n}{2})) \in O(n^2)$.

ALGO3 Für den dritten Algorithmus gilt:
Die äußere Schleife wird $\sqrt{n}$ mal betreten, was aus der Äquivalenzumformung von $i \cdot i < n$ folgt.
Die innere Schleife wird $\log_2 n$ mal betreten, was aus der Halbierung von j, das mit n initialisiert wird, bei jeder Iteration folgt.
Daraus folgt eine gesamte Laufzeit von $\sqrt{n} \cdot \log_2 n$.

3.2

$$T_(n) := \begin{cases} c_1, & \text{fuer } n < 4 \\ 2 \cdot T_(\frac{n}{4}) + 2 \cdot c_2 \cdot c_3 \cdot n + c_4, & \text{sonst} \end{cases}$$

Die Abbruchbedingung fuer die Rekursion ist eine Arraylaenge < 4, bei der die Funktion zudem keine von der Arraylaenge bzw. von n abhaengigen Berechnungen mehr ausfuehrt. Daraus ergibt sich: c_1 fuer $n < 4$.

In allen anderen Faellen werden einzelne Zuweisungsoperationen wie sum = 0, x = A.laenge/4 oder r = y mit konstantem Aufwand ausgefuehrt, was zu $+c_4$ in der Rekurrenzgleichung fuehrt.

Ausserdem werden unabhaengig davon und unabhaengig voneinander zwei for-Schleifen, deren Ruempfe ebenfalls einen konstanten Aufwand haben, jeweils n mal aufgerufen. Deswegen kommt der Summand $2 \cdot c_2 \cdot c_3 \cdot n$ hinzu.

Insbesondere wird auch noch 2 mal auf jeweils einem Viertel des urspruenglichen Arrays die Funktion erneut aufgerufen, weswegen als entscheidender Summand noch $2 \cdot T_(\frac{n}{4})$ addiert werden muss.

3.3

$$T_1(n) := \begin{cases} c_1, & \text{für } n = 1 \\ 8 \cdot T_1(\frac{n}{2}) + d_1 \cdot n^3, & \text{sonst} \end{cases}$$

mit $a = 8$ und $b = 2$ ist $\log_b(a) = 3$ und $f(n) = d_1 \cdot n^3$
Unter Verwendung des 1. Falles des Mastertheorems gilt:
$T(n) \in \Theta(n^{log_b(a)})$, falls $f(n) \in O(n^{log_b(a)-\epsilon})$ für ein $\epsilon > 0$.
Dies gilt hier leider nicht, da für kein $\epsilon > 0$ die Ungleichung $d_1 \cdot n^3 \leq n^{3-\epsilon}$ erfüllt ist, beziehungsweise nur für bestimmte d_1.
Betrachtung des 2. Falles des Mastertheorems:
Es gilt: $T(n) \in \Theta(n^{log_b(a)} \cdot log_2(n))$, falls $f(n) \in \Theta(n^{log_b(a)})$
Da für $\lim_{n\to\infty} \frac{f(n)}{n^{log_b(a)}} = \lim_{n\to\infty} \frac{d_1 \cdot n^3}{n^3} = \lim_{n\to\infty} \frac{d_1 \cdot 1}{1} = d_1$ die Beziehung $f(n) \in \Theta(n^{log_b(a)})$ erfüllt ist. Damit gilt $T_1(n) \in \Theta(n^{log_b(a)} \cdot log_2(n))$.

$$T_2(n) := \begin{cases} c_2, & \text{für } n = 1 \\ 5 \cdot T_2(\frac{n}{4}) + d_2 \cdot n^2, & \text{sonst} \end{cases}$$

mit $a = 5$ und $b = 4$ ist $log_b(a) \approx 1,16$ und $f(n) = d_2 \cdot n^2$
Unter Betrachtung des 3.Falles des Mastertheorems gilt:
$T(n) \in \Theta(f(n))$, falls $f(n) \in \Omega(n^{log_b(a)+\epsilon})$ für ein $\epsilon > 0$ und $a \cdot f(\frac{n}{b}) \leq \delta \cdot f(n)$ für ein $\delta < 1$ und große n.
Mit $\epsilon = 0,5$:
$\lim_{n\to\infty} \frac{f(n)}{n^{log_b(a)+\epsilon}} = \lim_{n\to\infty} \frac{d_2 \cdot n^2}{n^{1,16+0,5}} = \lim_{n\to\infty} \frac{d_2 \cdot n^2}{n^{1,66}} = \lim_{n\to\infty} \frac{d_2 \cdot n^{0,34}}{1} = \infty$
Hiermit ist $f(n) \in \Omega(n^{log_b(a)+\epsilon})$.

3.4