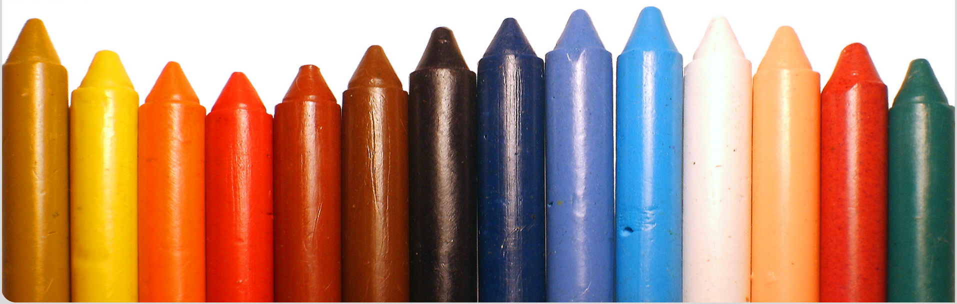
Algorithmen I - Sommersemester 2014

Tutorium Nr. 2

Tobias Hornberger | 1. Mai 2014

INSTITUT FÜR THEORETISCHE INFORMATIK



Rückgabe der Übungsblätter

$\rightarrow$ Bei Rückfragen bitte am Ende des Tutoriums vorkommen

Email-Liste

Nach dem Email-Desaster:

Kommunikation

- Bitte Email auf dem Blatt eintragen
- Nach dem Tutorium schauen ob meine Email angekommen ist

Rekurrenz Beispiel

$$T(n) = \begin{cases} 1, & \text{falls } n = 1 \\ 2T(\lfloor n/2 \rfloor) + n, & \text{falls } n \geq 2 \end{cases}$$

$$n \in 2^m \text{ für } m \in \mathbb{N}_0$$

Vorgehen

1. Lösung raten...
2. Beweisen, dass die Lösung stimmt

Umformung:
$T(2^m) = 2 * T(2^{m-1}) + 2^m$ mit $T(1) = 1$
$S(m) = 2 * S(m) + 2^m$ mit $S(0) = 1$

Ersten Werte:

- $S(0) = T(2^0) = T(1) = 1$
- $S(1) = T(2^1) = T(2) = 2 * 1 + 2 = 4$
- $S(2) = T(2^2) = T(4) = 2 * (2 * 1 + 2) + 4 = 12$
- $S(3) = T(2^3) = T(8) = 2 * (2 * (2 * 1 + 2) + 4) + 8 = 32$
- $S(4) = T(2^4) = T(16) = 2 * (2 * (2 * (2 * 1 + 2) + 4) + 8) + 16 = 70$

Rekurrenzen

Durch scharfes Hinsehen erkennen wir eine mögliche Lösung:

$$S(m) = 2^{m+1} + (m-1) * 2^m$$

Beweis der Lösung durch völlständige Induktion.

Beweis durch vollständige Induktion

IA: $S(0) = T(2^0) = T(1) = 1$
IV: $S(m) = 2^{m+1} + (m-1) * 2^m$
IS: m $\rightarrow m+1$

$$
\begin{aligned}
S(m+1) &= 2 * S(m) + 2^{m+1} && (1)\\
&= 2 * (2^{m+1} + (m-1) * 2^m) + 2^{m+1} && (2)\\
&= 2^{m+2} + (m-1) * 2^{m+1} + 2^{m+1} && (3)\\
&= 2^{m+2} + m * 2^{m+1} && (4)
\end{aligned}
$$

Zurück nach T(n) rechnen für Finale Lösung:
$T(n) = S(\log_2 n) = 2^{\log_2 n+1} * (\log_2 n - 1) * 2^{\log_2 n}$

Test Test

Test