DAP2 – Präsenzübung 3

Besprechung: 10.05.2017 — 12.05.2017

Präsenzaufgabe 3.1: (Rekursionsgleichung)

Gegeben sei die Rekursionsgleichung:

$$T(n) = \begin{cases} 4 \cdot T(n/2) + n^3 & \text{wenn } n > 1 \\ 1 & \text{sonst} \end{cases}$$

a) Bestimmen Sie eine asymptotische obere Schranke für $T(n)$ und beweisen Sie sie mittels Induktion. Sie dürfen annehmen, dass n von der Form 2^k für ein $k \in \mathbb{N}$ ist.

b) Begründen Sie, warum die Annahme über n keine Beschränkung der Allgemeinheit darstellt.

Präsenzaufgabe 3.2: (Teile und Herrsche)

Gegeben sei ein Array $A[1..n]$ von positiven ganzen Zahlen. Wir wollen die größte Wertdifferenz $A[i]-A[j]$ für $i \leq j$ bestimmen. Z.B. im Array $\{1, 8, 4, 7\}$ ist die größte Wertdifferenz $8-4=4$.

a) Entwickeln Sie einen Teile-und-Herrsche Algorithmus, der die größte Wertdifferenz berechnet und zurückgibt, und beschreiben Sie ihn mit eigenen Worten. Geben Sie eine Implementierung Ihres Algorithmus in Pseudocode an. Für die volle Punktzahl wird ein Algorithmus erwartet, dessen Worst-Case-Laufzeit durch $\mathcal{O}(n)$ beschränkt ist.

b) Analysieren Sie die Laufzeit Ihres Algorithmus. Stellen Sie hierzu eine Rekursionsgleichung für die Laufzeit Ihres Algorithmus auf und lösen Sie diese. Hier dürfen Sie behaupten, dass n eine Zweierpotenz ist.

c) Zeigen Sie die Korrektheit Ihres Algorithmus.