

DAP2 – Präsenzübung 6

Besprechung: 31.05.2017 — 02.06.2017

Präsenzaufgabe 6.1: (Dynamische Programmierung)

In der Vorlesung wurde ein dynamisches Programm zur Maximumssuche besprochen. Des Weiteren wurde eine Idee angegeben, wie man den Index eines größten Elements in einem Array A der Länge n dynamisch bestimmen kann (siehe Foliensatz 8, Folie 58):

Sei

$$\text{Max}(i) = \begin{cases} -\infty & \text{falls } i = 0 \\ A[1] & \text{falls } i = 1 \\ \max\{\text{Max}(i-1), A[i]\} & \text{sonst} \end{cases}$$

die rekursive Funktion, die für ein i , $1 \leq i \leq n$, das Maximum $\max_{i \leq j \leq n} \{A[j]\}$ des Arrays bis zur Stelle i berechnet.

Der Index j des maximalen Elements ist der größte Wert, für den $\text{Max}(j) > \text{Max}(j-1)$ gilt.

Untersuchen Sie dies wie folgt.

- Geben Sie ein dynamisches Programm in Pseudocode an, das basierend auf diesen Überlegungen einen Index eines größten Elements in einem Array ausgibt.
- Analysieren Sie die Laufzeit Ihres Algorithmus.
- Beweisen Sie die Aussage, dass der größte Wert j mit $\text{Max}(j) > \text{Max}(j-1)$ der gesuchte Index des Maximums ist.

Präsenzaufgabe 6.2: (Dynamische Programmierung)

Betrachten Sie das folgende Partitionierungs-Problem: Gegeben eine Menge $A = \{a_1, \dots, a_m\}$ bestehend aus natürlichen Zahlen, gibt es eine Aufteilung der Zahlen in drei Teilmengen, sodass die Summe der Elemente jeweils gleich groß ist?

- Formulieren Sie eine geeignete rekursive Form für dieses Problem.
- Geben Sie einen Algorithmus in Pseudocode an, der das Gesamtproblem basierend auf der rekursiven Beziehung aus Teilaufgabe a) mit dynamischer Programmierung löst.
- Analysieren Sie die Laufzeit Ihres Algorithmus.
- Beweisen Sie die Korrektheit der in Teilaufgabe a) angegebenen rekursiven Form.