

Algorithmen und Datenstrukturen
SS 2019

Übungsblatt 13: Paradigmen II

Aufgabe 13-1 *Dynamische Programmierung*

- (a) Berechnen Sie die Edit-Distanz der Wörter ALGORITHMEN und DATENSTRUKTUREN mit Hilfe dynamischer Programmierung.
- (b) Geben Sie die Komplexität der dynamischen Programmierung für dieses Beispiel im Gegensatz zu einer naiven rekursiven Lösung zur Berechnung der Edit-Distanz an.

Aufgabe 13-2 *Greedy Algorithmen, Heuristiken und dynamische Programmierung*

In der Vorlesung haben Sie kennengelernt, dass Greedy Algorithmen, welche auf Heuristiken basieren, oft nicht zur optimalen Lösung von Problemen führen. Dennoch finden sich in der Industrie sehr viele Beispiele, bei denen Greedy Algorithmen zur Approximation einer optimalen Lösung eingesetzt werden.

- (a) Greedy Algorithmen können die optimale Lösung finden, tun es aber nicht immer. Warum werden überhaupt Greedy Algorithmen verwendet und nicht stattdessen Algorithmen, welche eine optimale Lösung garantieren?
- (b) Das Rucksackproblem gehört zur Liste der 21 klassischen NP-vollständigen Problemen. Um eine optimale Lösung für dieses Problem finden zu können, gibt es bisher nur den Lösungsansatz alle Möglichkeiten systematisch durchzuprobieren.
 - (i) Welche Komplexität ergibt sich demnach für den Algorithmus, welcher die optimale Lösung des Rucksackproblems mit n verschiedenen Gegenständen liefert.
 - (ii) Wie groß ist die Komplexität, wenn eine optimale Lösung mit Hilfe von der in der Vorlesung besprochenen dynamischen Programmierung für das Rucksackproblem mit n verschiedenen Gegenständen und einer Kapazität von b berechnet wird.
 - (iii) Warum ist das Ergebnis von (ii) kein Widerspruch zu (i)?