

Algorithmen und Datenstrukturen

Übungszettel 8

Martin Avanzini <martin.avanzini@uibk.ac.at>
Thomas Bauereiß <thomas.bauereiss@uibk.ac.at>
Herbert Jordan <herbert@dps.uibk.ac.at>
René Thiemann <rene.thiemann@uibk.ac.at>

17. Mai, zur Besprechung am 24. Mai

Aufgabe 1) Dynamische Programmierung Ein Tourist möchte aus seinem Urlaub eine Sammlung an Kräutern mit nachhause nehmen. Leider erlaubt ihm die Fluglinie höchstens 20kg an Gepäck - von denen bereits 15kg für Kleidung verbraucht werden. In der nachfolgenden Tabelle finden sich die Masse und der geschätzte ideelle Wert diverser Optionen:

Sorte	A	B	C	D	E	F
Masse [kg]	0.5	2	1	1.5	1.5	0.5
Wert [GE]	250	400	300	350	280	240

Welche Sorten soll der Tourist zur Auffüllung der restlichen 5kg in seinen Koffer packen um den größten Wert zu erzielen? Hierbei können Päckchen nicht geteilt werden und stehen höchstens 1-mal zur Verfügung.

1. Geben Sie für den allgemeinen Fall (beliebige Anzahl an Sorten n , beliebige Gewichtsbeschränkung b) eine Rekursionsgleichung an, die den maximalen Ertrag e liefert. Die Massen und die Werte der einzelnen Sorten sind in den Arrays m und v gespeichert. Geben Sie präzise an, was $e(\dots)$ ausdrückt (Schritte 1+2 auf Folie 259).
2. Implementieren Sie den resultierenden Algorithmus in Java und geben Sie eine optimale Lösung des oben beschriebenen, speziellen Problems an (Schritte 3+4 auf Folie 259). Wählen Sie hierfür eine geeignete Datenstruktur zur Darstellung der Sortenauswahl.
3. Geben Sie die Laufzeit Ihrer Lösung an.

Aufgabe 2) Greedy-Verfahren: Huffman Codierung In der Vorlesung wurde die Huffman Codierung als ein Beispiel für einen Greedy Algorithmus beschrieben.

1. Fassen Sie den Algorithmus zur Huffman Codierung möglichst kompakt (und exakt) mittels Pseudocode zusammen (Folien 284-290). Beschreiben Sie ebenfalls die notwendigen Ein- und Ausgaben.

2. Worin besteht das "greedy" bei diesem Algorithmus? Was ist eine *gierige* Entscheidung?
3. Codieren Sie die Zeichenkette "hello world" mittels der Huffman Codierung. Geben Sie hierfür das Alphabet, die Häufigkeiten, die Erstellung sowie den resultierenden Prefix-Code sowie die Länge der komprimierten sowie unkomprimierten Zeichenkette an. Um identische Ergebnisse zu erhalten sollten bei gleicher Häufigkeit von Symbolen die alphabetisch kleineren bevorzugt werden.

Aufgabe 3) Entwicklung eines Greedy-Verfahren Ein typisches Problem aus dem Supermarkt: ein Betrag soll mit Wechselgeld zusammengestellt werden. Hierbei soll eine möglichst geringe Anzahl an Münzen verwendet werden.

Sei $W = \{w_1, w_2, \dots, w_n\}$ mit $w_1 = 0.01$ und $w_j < w_{j+1}$ für alle $1 \leq j < n$ die Menge der zur Verfügung stehenden Münzbeträge. Die Rekursionsgleichung für die Gesamtanzahl an Münzen $c(b)$ für den Betrag b kann wie folgt formuliert werden:

$$c(b) = \begin{cases} 0 & b = 0 \\ \min \{1 + c(b - k) \mid k \in W \wedge k \leq b\} & b > 0 \end{cases}$$

1. Formulieren Sie ein Kriterium für eine gierige Auswahl zur Lösung des Optimierungsproblems entsprechend dem 3 Schritte Verfahren von Folie 283.
2. Demonstrieren Sie den aus Ihrem Kriterium resultierenden Algorithmus indem Sie das Wechselgeld für einen Betrag von €1.48 berechnen. Die zur Verfügung stehenden Münzen sind die üblichen 1, 2, 5, 10, 20 und 50 Cent Stücke sowie 1 und 2 Euro Münzen.
3. Liefert Ihr Greedy-Algorithmus immer ein Optimum für a) Euro-Beträge mit den gängigen Münzen bzw. b) für eine beliebige Menge an Münzbeträgen W ? Untermauern Sie Ihre Behauptung mit einer entsprechenden Beweis bzw. einem Gegenbeispiel.