

Aufgabe 4

Das GUTSCHEIN-Problem ist gegeben durch eine Folge w_1, \dots, w_n von Warenwerten (wobei $w \in \mathbb{N}_0$ für $i = 1, \dots, n$) und einem Gutscheinbetrag $G \in \mathbb{N}_0$.

Da Gutscheine nicht in Bargeld ausgezahlt werden können, ist die Frage, ob man eine Teilfolge der Waren findet, sodass man genau den Gutschein ausnutzt. Formal ist dies die Frage, ob es eine Menge von Indizes I mit $I \subseteq \{1, \dots, n\}$ gibt, sodass $\sum_{i \in I} w_i = G$

(a) Sei $w_1 = 10, w_2 = 30, w_3 = 40, w_4 = 20, w_5 = 15$ eine Folge von Warenwerten.

(i) Geben Sie einen Gutscheinbetrag $40 < G < 115$ an, sodass die GUTSCHEIN-Instanz eine Lösung hat. Geben Sie auch die lösende Menge $I \subseteq \{1, 2, 3, 4, 5\}$ von Indizes an.

50
 $I = \{1, 3\}$

(ii) Geben Sie einen Gutscheinbetrag G mit $40 < G < 115$ an, sodass die GUTSCHEIN-Instanz keine Lösung hat.

51

(b) Sei $table$ eine $(n \times (G + 1))$ -Tabelle mit Einträgen $table[i, k]$, für $1 < i < n$ und $0 \leq k \leq G$, sodass

$$table[i, k] = \begin{cases} \text{true} & \text{falls es } I \subseteq \{1, \dots, n\} \text{ mit } \sum_{i \in I} w_i = G \text{ gibt} \\ \text{false} & \text{sonst} \end{cases}$$

Geben Sie einen Algorithmus in Pseudo-Code oder Java an, der die Tabelle $table$ mit *dynamischer Programmierung* in Worst-Case-Laufzeit $\mathcal{O}(n \times G)$ erzeugt. Begründen Sie die Korrektheit und die Laufzeit Ihres Algorithmus. Welcher Eintrag in $table$ löst das GUTSCHEIN-Problem?