PD Dr. Elmar Langetepe
David Kübel
Institut für Informatik

Übungsblatt 5

Aufgabe 5.1: (2+6 Punkte)

(a) Wir betrachten die Instanz des Rucksackproblems mit 4 Objekten $i = 1, \ldots, 4$, die durch die folgende Tabelle gegeben ist.

i	1	2	3	4
w_i	3	1	2	5
g_i	2	4	3	6

Stellen Sie die Tabelle mit den Werten $W(i, g)$, die durch den Algorithmus berechnet werden, auf und geben Sie eine optimale Lösung für $G = 7$ und $G = 10$ an.

(b) Wir haben gesehen, wie mit dynamischer Programmierung optimale Lösungen des Rucksackproblems mit einer Laufzeit von $O(n \cdot G)$ berechnet werden können, wobei n die Anzahl der Objekte ist. Zeigen Sie, wie eine optimale Lösung des Rucksackproblems mit einer Laufzeit von $O(n^2 M)$ berechnet werden kann, wobei

- $M = \max_{i \in \{1,\ldots,n\}} g_i$, das maximale Gewicht eines Objekts bzw.
- $M = \max_{i \in \{1,\ldots,n\}} w_i$, den maximalen Wert eines Objekts

bezeichnet.

Aufgabe 5.2: (4 Punkte)

Wir betrachten das ganzzahlige Rucksackproblem eingeschränkt auf Instanzen, in denen die Gewichte aufsteigend und die Nutzen absteigend sortiert sind, d. h. $w_1 \leq \ldots \leq w_n$ und $p_1 \geq \ldots \geq p_n$. Geben Sie an, wie diese Variante des Rucksackproblems mithilfe eines Greedy-Algorithmus optimal gelöst werden kann.

Aufgabe 5.3: (4+4 Punkte)

Der Kaffeeautomat im Informatikgebäude bietet Heißgetränke zu verschiedenen Preisen an. Um nach einem Kauf Wechselgeld der Höhe x auszugeben, stehen dem Automat Münzen mit den Werten $c_1, \ldots, c_k$ zur Verfügung, wobei jeweils $c_i < c_{i+1}$ gilt. In Euro z.B.: 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2.

1. Beschreiben Sie einen Greedy-Algorithmus, der das Wechselgeld in Euro erstattet. Aus praktischen Gründen soll die Anzahl zurückgegebener Münzen so klein wie möglich sein. Nehmen Sie an, dass jeder Münzwert ausreichend oft vorhanden ist.
2. Geben Sie ein Beispiel für eine (fiktive) Währung an, bei welcher der Greedy-Algorithmus kein optimales Ergebnis liefert. Nennen Sie eine Bedingung, die an eine Währung mit $k \geq 3$ unterschiedlichen Münzwerten gestellt werden kann, sodass das Greedy Verfahren immer ein optimales Ergebnis liefert! Begründen Sie, warum Ihre Bedingung dies garantiert.

Aufgabe 5.4: (4 Punkte)

Gegeben sei eine Menge $I = \{i_1, \ldots i_n\}$ von Intervallen, d. h. $i_k = (a_k, b_k)$ mit $a_k, b_k \in \mathbb{R}$ für $1 \leq k \leq n$. Geben Sie einen *Sweep*-Algorithmus an, der mit Laufzeit $O(n \log n)$ und linearem Speicherplatz überprüft, ob sich zwei der Intervalle überlappen.

Aufgabe 5.5: (6 Zusatzpunkte)

Wir betrachten das sogenannte *Partitionsproblem*: Gegeben seien natürliche Zahlen $a_1, \ldots, a_n$. Es soll entschieden werden, ob es eine Indexmenge $I \subseteq \{1, \ldots, n\}$ gibt, sodass $\sum_{i \in I} a_i = \sum_{i \in \{1,\ldots,n\} \setminus I} a_i$ gilt, und gegebenenfalls eine solche Indexmenge zu finden.

Benutzen sie dynamische Programmierung, um das Partitionsproblem mit einer Laufzeit von $O(nS)$ zu lösen, wobei $S = \sum_{i=1}^{n} a_i$ ist.

Hinweis: Eine Indexmenge I besitzt genau dann die gewünschte Eigenschaft, wenn $\sum_{i \in I} a_i = \frac{S}{2}$ gilt.