

Algorithmen und Datenstrukturen

Aufgabenblatt 02

Abgabe bis: 06.05.2021 22:00 Uhr

Bitte bearbeiten Sie die Aufgaben in Dreiergruppen.

1. Laufzeitverhalten, Implementierung

Implementieren Sie die beiden Rucksack-Algorithmen aus der Vorlesung in Java und in Python. Messen Sie wieder die Zeit und zählen Sie, wie oft die Funktion **max** aufgerufen wird, bzw. wie oft die innere Schleife durchlaufen wird. Halten Sie Ihre Ergebnisse in einer Tabelle fest. Nehmen Sie dabei einmal die Eingabedaten aus der Vorlesung und erstellen zusätzlich einen zweiten Datensatz mit 20 Gegenständen als Eingabe. Interpretieren Sie Ihre Ergebnisse.

Ziel: Programmierstrategien

2. Greedy

Implementieren Sie den in der Vorlesung vorgestellten Algorithmus zur Rückgabe von Wechselgeld mit dem Greedy-Verfahren (in Java oder Python). Hierbei soll Ihnen jedoch nicht unendlich viel Kleingeld zur Verfügung stehen, sondern Sie haben eine Kasse mit einer bestimmten Anzahl Münzen von jeder Sorte.

Eingabe: Über Tastatur:

Die zu zahlende Summe $S \in \mathbb{N}$,

der gegebene Betrag $B \in \mathbb{N}$,

die verfügbaren Münzgrößen $m_i, 1 \leq i \leq n$,

die vorhanden Anzahlen $a_i, 1 \leq i \leq n$, jeder Münzart

Ausgabe: das Wechselgeld in Form (m_j, w_j) ,

m_j = Münzgröße,

w_j = Anzahl der von der Münzgröße m_j zurückgegebenen Münzen

“Geldwechseln nicht möglich”, falls nicht genug Wechselgeld da ist.

Überlegen Sie, welche Datenstrukturen Sie am besten wo benutzen. Begründen Sie Ihre Wahl der eingesetzten Datenstrukturen. Überlegen Sie sich zwei Kombinationen vorhandener Münzen, in denen der Algorithmus das Optimum liefert und zwei, in denen er schlecht arbeitet. (Das Beispiel aus der Vorlesung zählt nicht.) Sehen Sie vor, dass man bei einem Programmlauf mehrmals Wechselgeld ausgeben kann, ohne die Eingangswerte alle noch einmal eintippen zu müssen.

Ziel: Implementierung eines Greedy-Algorithmus.

3. Backtracking

Schreiben Sie ein Programm, das das 4-Damenproblem mit Hilfe von Backtracking löst. Lässt sich Ihre Lösung auf n Damen erweitern?

Ziel: Implementierung von Backtracking.

Aufgabe 1)

Aus unseren Ergebnissen lässt sich zum einen Interpretieren, dass die Backtracking-Variante des Algorithmusses ein schlechteres Laufzeitverhalten hat, zum anderen dass die Java-Implementierung deutlich schneller ist als die Python-Implementierung.