

Aufgabe 1 Bottom-up Merge Sort

10 Punkte

In der Vorlesung haben Sie eine rekursive Variante von Merge Sort gesehen. Entwerfen Sie eine iterative Variante von Merge Sort und testen Sie diese.

Aufgabe 2 Anwendungen von Sortieralgorithmen

10 Punkte

Um Ihre finanzielle Situation zu verbessern, haben Hannelore und Werner eine Stelle im Supermarkt angetreten.

- (a) Im Lager befindet sich ein Haufen von n Mehlsäcken, die alle paarweise ein unterschiedliches Gewicht besitzen. Werner erhält die Aufgabe, herauszufinden, ob es zwei Mehlsäcke gibt, deren Gewicht sich nur um 10 Gramm unterscheidet. Dazu erhält er eine Balkenwaage, mit der zwei Mehlsäcke vergleichen kann. Dabei lässt sich herausfinden, welcher der beiden Säcke schwerer ist, und ob sich das Gewicht um höchstens 10 Gramm unterschreidet.

Beschreiben Sie einen möglichst schnellen Algorithmus, der das Problem löst, und implementieren Sie Ihren Algorithmus in Python (Sie können die Mehlsäcke als eine Liste von Zahlen modellieren).

Bemerkung. “Möglichst schnell” bedeutet: Der Algorithmus sollte nicht alle Paare von Mehlsäcken vergleichen.

- (b) Hannelore arbeitet derweil an der Käsetheke. Dort gibt es einen Stapel mit Gouda-Packungen, einen Stapel mit Emmentaler-Packungen und einen Stapel mit Roquefort-Packungen. Die einzelnen Packungen können unterschiedliche Gewichte haben. Hannelores Cousine Ursel tritt an die Käsetheke, und die beiden unterhalten sich über den neuesten Durchbruch aus der Quantenkomplexitätstheorie ($RE = QMIP$).

Dann bestellt Ursel eine Packung Gouda, eine Packung Emmentaler, und eine Packung Roquefort, so dass die drei Packungen zusammen genau 1234 Gramm wiegen sollen. Beschreiben Sie einen möglichst schnellen Algorithmus, der drei solche Packungen findet, falls es sie gibt, und implementieren Sie Ihren Algorithmus in Python. (Sie können die Käsepackungen als drei Listen von Zahlen modellieren).

Bemerkung. “Möglichst schnell” bedeutet: Der Algorithmus sollte nicht alle Tripel von Käsepackungen vergleichen.

Aufgabe 3 Stabiles Sortieren

10 Punkte

Das *stabile Sortierproblem* ist folgendermaßen gegeben:

Eingabe: Eine Liste x_1, x_2, \dots, x_n mit n Elementen aus einer total geordneten Menge.

Ausgabe: Eine Permutation $\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$, so dass gilt: (i) $x_{\sigma(1)} \leq x_{\sigma(2)} \leq \dots \leq x_{\sigma(n)}$ und (ii) für alle $1 \leq i < n$: wenn $x_{\sigma(i)} = x_{\sigma(i+1)}$ ist, dann ist $\sigma(i) < \sigma(i+1)$.

Ein Sortierverfahren heißt *stabil*, falls es das stabile Sortierproblem löst.

- (a) Beschreiben Sie in zwei Sätzen, was es heißt, dass ein Sortierverfahren stabil ist, und geben Sie zwei Beispiele, eines, bei dem eine Sortierung stabil ist und eines, bei dem eine Sortierung nicht stabil ist.
- (b) Bestimmen Sie für die folgenden Sortierverfahren aus der Vorlesung, ob sie stabil sind oder nicht: Selection Sort, Insertion Sort, Quick Sort, und Merge Sort.

Geben Sie jeweils eine kurze Begründung.

- (c) Beschreiben Sie einen einfachen Weg, wie sich *jedes* vergleichsbasierte Sortierverfahren zu einem stabilen Sortierverfahren überführen lässt.