

Algorithmen und Datenstrukturen INF3/ICS3

Wintersemester 2023/24

Prof. Dr. Georg Schied

Aufgabenblatt 6

Abgabetermin: Mo. 27. November 2023, 23:59 Uhr
Zum Bestehen müssen 10 von 20 Punkten erreicht werden.

Aufgabe 6.1 - Scheinaufgabe (5 P)

Es soll folgendes sortiert werden:

- a) Bücher bei Amazon nach Preis
- b) Bücher nach 10-stelliger ISBN-Nummer
- c) Gemeinden Deutschlands nach Anzahl der Einwohner
- d) Briefe nach Postleitzahl
- e) Historische Persönlichkeiten nach Geburtsdatum (Jahr/Monat/Tag)
- f) Die Masse von physikalischen Objekten aller Art (z.B. Elektronen, Autos, Planeten, Galaxien, ...).

Wäre **Bucketsort** gut dazu geeignet? Geben Sie jeweils eine kurze Begründung.

Aufgabe 6.2

Wäre **Radixsort** gut dazu geeignet, die in der vorigen Aufgabe angegebenen Arten von Daten zu sortieren? Geben Sie jeweils eine kurze Begründung.

Aufgabe 6.3 - Scheinaufgabe (5 P)

Sortieren durch Fachverteilung (bucketsort) soll verwendet werden, um Artikel nach Artikelnummern zu sortieren. Es sei vorausgesetzt, dass die Artikelnummern immer im Bereich von 0 bis 999 liegen. Es ist außerdem die Konstante $c = 3$ vorgegeben.

- a) Zeigen Sie, wie folgende 15 Artikelnummern mittels Bucketsort sortiert werden:

970, 284, 98, 665, 535, 466, 134, 558, 281, 18, 648, 198, 92, 597, 622

Geben Sie die wesentlichen Zwischenschritte an, damit die Vorgehensweise nachvollziehbar ist.

- b) Was würde sich an der Vorgehensweise ändern, wenn nicht 15, sondern 300 Artikelnummern zu sortieren wären?

Aufgabe 6.4

Zeigen Sie, wie folgende Zeichenketten über dem Alphabet $\Sigma = \{a, b, c, d, e\}$ mittels **Radixsort** sortiert werden.

baad, abcd, edbb, abaa, ccdd, ddee, bedc, eeca, bcda, eaae

Aufgabe 6.5 - Scheinaufgabe (10 P)

- a) In Moodle ist die Klasse `Document` gegeben, die ein Dokument mit einer Dokumenten-ID und einem Titel repräsentiert. Eine Dokumenten-ID ist dabei immer eine Folge von 8 Dezimalziffern (als `String`, führende Nullen erlaubt). Implementieren Sie in Klasse `Document` eine Methode

```
public static void documentSort(Document[] dlist),
```

die ein Array von `Document`-Objekten mittels **Radixsort** nach der Dokumenten-ID sortiert. In Moodle finden Sie dazu auch eine JUnit-Testklasse.

Tipp: Sie können dazu z.B. die Klasse `ArrayList` oder `LinkedList` aus der Java-Standardbibliothek (Paket `java.util`) verwenden.

- b) Messen Sie die Laufzeiten zum Sortieren von $n = 10, 100, \dots, 1$ Mio. Dokumenten mittels der Klasse `DocumentSortMeasurement`, die in Moodle auch gegeben ist (mit Option `-Xint`). Passen die gemessenen Laufzeiten zu den theoretisch erwarteten Werten?

Es werden zum Vergleich dabei auch die Laufzeiten für das Sortieren mit `Arrays.sort()` aus der Java-Standardbibliothek bestimmt. Schaffen Sie es, mit Ihrer Radixsort-Implementierung schneller zu sein?

Aufgabe 6.6

Erweitern Sie die in der Vorlesung vorgestellte algebraische Spezifikation für den Datentyp **Nat** (natürliche Zahlen) um eine Operation *mult*, die das Produkt von zwei Zahlen berechnet.