

# Algorithmen & Datenstrukturen

## Blatt 2

Dr. Matthias Thimm

Tina Walber, Leon Kastler, Martin Leinberger und Maximilian Strauch

Fachbereich Informatik, Universität Koblenz-Landau

7. Dezember 2013

## 1 Rekursiv zu iterativ (4 Punkte)

Schreiben Sie den folgenden rekursiven Algorithmus in einen iterativen Algorithmus um:

```
static long calculateRecursive(long x, long y) {
    return calculateRecursive(x, y, 0);
}

static private long calculateRecursive(long x, long y, long result) {
    if(x > 0) {
        return calculateRecursive(x-1, y, result + x * y);
    }
    else {
        return result;
    }
}
```



## 2 Verständnisfragen (4 Punkte, je 0.5)

Beantworten Sie folgende Fragen zu Datenstrukturen kurz und präzise:

- Erklären Sie den Unterschied zwischen Listen und Bäumen.
- Erklären Sie den Unterschied zwischen Bäume und Graphen.
- Kann ich eine Liste als Baum darstellen? Falls ja, wie?
- Kann ich einen Baum als Liste darstellen? Falls ja, wie?
- Welche lineare Datenstruktur würden Sie verwenden, wenn Sie die Fibonacci-Zahlen von 1 bis 20 ausrechnen wollen? Begründen Sie Ihre Antwort.
- Welche lineare Datenstruktur würden Sie verwenden, wenn Sie alle Studierenden der Universität Koblenz auflisten wollen, deren Nachname mit A, C oder F beginnt? Begründen Sie Ihre Antwort.
- Welche Eigenschaft muss eine Folge haben, damit die binäre Suche angewendet werden kann?
- Wann ist die sequenzielle Suche effizienter als die binäre Suche?



### 3 Vergleichsverfahren (4 Punkte)

Julia und Christoph sammeln beide Euromünzen. Julia hat  $n$  Münzen und Christoph  $m$ . Sie müssen eine Münze in die Parkuhr werfen und wollen dafür eine Münze aussuchen, die sie beide besitzen. Zudem wollen sie nur Münzen ansehen, die in Münzwert und Herkunftsland übereinstimmen.

1. Christoph schlägt Verfahren 1 vor: Julia wählt eine Münze ihrer Sammlung aus. Dann vergleicht sie diese der Reihe nach mit allen Münzen in Christophs Sammlung. Sie vergleicht die ersten beiden Münzen, sind sie gleich, ist das Verfahren erfolgreich beendet. Sind die Münzen unterschiedlich fährt sie mit der nächsten von Christophs Münzen fort usw. Ist Julias Münze mit allen aus Christophs Sammlung verglichen worden, wird das Verfahren mit der nächsten Münze aus ihrer Sammlung fortgesetzt.
2. Julia schlägt Verfahren 2 vor: Christoph und Julia sortieren ihre Münzsammlungen, zuerst nach aufsteigender Wertigkeit und dann nach einer bestimmten Länderreihenfolge. Sie vergleichen dann jeweils ihre geringwertigsten Münzen. Sind beide gleich, ist das Verfahren erfolgreich beendet. Sind sie unterschiedlich, wird die nach dem Sortierkriterium kleinere der beiden zur Seite genommen und das Verfahren ohne diese Münze entsprechend fortgesetzt.

Beantworten Sie die folgenden Fragen, Begründungen angeben:

- Findet Verfahren 1 zwei gleiche Münzen, wenn es diese gibt?
- Findet Verfahren 2 zwei gleiche Münzen, wenn es diese gibt?
- Wie viele Vergleiche werden bei Verfahren 1 im besten und im schlechtesten Fall durchgeführt?
- Wie viele Vergleiche werden bei Verfahren 2 im besten und im schlechtesten Fall durchgeführt? (ohne Berücksichtigung des Sortieraufwands)



## 4 Fibonacci Suche (8 Punkte)

Entwickeln Sie einen Algorithmus, um in einer Liste von Namen möglichst schnell einen bestimmten Namen zu finden. In der Vorlesung wurde der Algorithmus zur Fibonacci Suche vorgestellt, dieser soll zur Lösung der Aufgabe genutzt werden. Gehen Sie wie folgt vor:

- Lesen Sie zuerst die Datei `namen.txt` ein und speichern diese in einem Array ab. Sie finden die Datei im SVN: <https://svn.uni-koblenz.de/west/aud-ws1314/allgroups/assignments/ass02/names.txt>
- Entwickeln Sie einen **iterativen** Algorithmus zur Fibonacci Suche um den gewünschten Vornamen vom Computer finden zu lassen.
- Schreiben Sie hierfür ein kleines Benutzerinterface das dem Benutzer zunächst einen Namen vorschlägt. Der Benutzer entscheidet dann mit Hilfe der Tasten „<“ und „>“ ob der Name, nach dem er oder sie sucht, im Alphabet früher („<“) oder später („>“) kommt. Daraufhin wird die Suche fortgesetzt, solange bis der richtige Namen gefunden wurde.

### Hinweis

Zum Einlesen der Tastaturbefehle können Sie z.B. die Klasse `java.util.Scanner`<sup>1</sup> benutzen. Zum Einlesen der Datei können Sie z.B. die Klasse `java.io.BufferedReader`<sup>2</sup> zusammen mit der Klasse `java.io.FileReader`<sup>3</sup> verwenden. Beachten Sie hierbei, dass die Kommentarseiten, welche mit einer Raute (`#`) anfangen, ignoriert werden müssen. Dublikate müssen nicht herausgefiltert werden.

---

<sup>1</sup><http://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html>

<sup>2</sup><http://docs.oracle.com/javase/7/docs/api/java/io/BufferedReader.html>

<sup>3</sup><http://docs.oracle.com/javase/7/docs/api/java/io/FileReader.html>

