



Algorithmische Mathematik I

Wintersemester 2017/18
Prof. Dr. Ira Neitzel
AR. Dr. Tino Ullrich



Übungsblatt 8.

Abgabe am **04.12.2017** vor der Vorlesung.

Aufgabe 1. (Die Türme von Hanoi)

Bei diesem Spiel gibt es drei Stäbe A, B und C . Auf den Stab A werden n verschieden große Scheiben der Größe nach gestapelt. Ziel ist es, diese Scheiben der Größe nach auf den Stab C umzusortieren. Es dürfen nur kleinere Scheiben auf größere gelegt werden und alle Scheiben müssen stets im Spiel bleiben. Dies geht natürlich nicht, ohne den Stab B zu Hilfe zu nehmen. Definieren Sie eine rekursive

```
1 Funktion bewege (Zahl n, Stab A, Stab B, Stab C)
```

die n Scheiben von Stab A (über Stab B) nach Stab C gemäss obiger Regeln verschiebt, und notieren Sie diese in Pseudocode. Geben Sie die Komplexität (Laufzeit) t_n dieses Algorithmus in der Landaunotation \mathcal{O} an.

Hinweis: Beginnen Sie mit $t_n = t_{n-1} + 1 + t_{n-1}$ und verwenden Sie Aufgabe 3 vom Blatt 1.

(3 + 3 = 6 Punkte)

Aufgabe 2. (Mergesort)

a. Gegeben sei folgende Zahlenfolge:

7, 17, 5, 3, 2, 11, 13, 1

Sortieren Sie diese Zahlenfolge mit dem Mergesort-Algorithmus!

Geben Sie die Aufteilung der Felder in jedem Schritt an. Markieren Sie alle Zahlen in einem Feld mit 1, 2, 3 4, 5, 6.

(6 Punkte)

Aufgabe 3. (Fibonacci II)

Analysieren Sie den Rechenaufwand zur Berechnung der n -ten Fibonaccizahl mittels

- der (naiven) rekursiven Definition $f_n = f_{n-1} + f_{n-2}$, $n \geq 2$, $f_0 = f_1 = 1$,
- durch Erzeugen eines Arrays von Fibonaccizahlen (siehe Programmieraufgabe 1b vom Blatt 7).

Hinweis: Verwenden Sie Aufgabe 2 vom Blatt 7 für a).

(2 + 2 = 4 Punkte)

Aufgabe 4. (Sortieren durch Einfügen)

Wir betrachten folgenden Sortieralgorithmus, welchen man oft beim Kartenspiel anwendet. Man zieht die Elemente eines (unsortierten) Arrays eines nach dem anderen und fügt jedes an seinem richtigen Platz zwischen den bereits betrachteten ein (wobei diese sortiert bleiben). Das betrachtete Element wird eingefügt, indem die größeren Elemente einfach um eine Position nach rechts bewegt werden. Notieren Sie diesen Algorithmus in Pseudocode. Was ist hinsichtlich der worst-case Komplexität zu erwarten?

(4 Punkte)

Programmieraufgabe 1. (Mergesort)

Implementieren Sie eine Sortierfunktion basierend auf dem Mergesort-Algorithmus, um eine Datenbank bestehend aus Studentendaten

```
1 struct Student {  
    long matnr;  
3    short geb[3];  
    short studienfach;  
5    short fachsemester;  
}
```

einmal nach Größe der Matrikelnummer und ein anderes Mal nach dem Geburtsdatum zu sortieren. Das Geburtsdatum ist dabei als dreielementiges Integerarray gegeben.

- Definieren Sie als erstes den `struct Student`. Die Matrikelnummer ist eine sechsstellige Zahl und das Geburtsdatum ein dreielementiges Array {Tag, Monat, Jahr} aus Integern. Studienfach und Fachsemester ist jeweils durch eine Zahl zwischen 1 und 100 kodiert.
- Erzeugen Sie anschließend eine (große) Datenbank mit beispielsweise $N = 10000$ zufälligen Studenten. Diese könnte man als Array (von `Student`) oder als verkettete Liste anlegen. Achten Sie darauf, dass Sie keinen 30. Februar o.ä. generieren.
- Überlegen Sie sich hinsichtlich des zusätzlichen Speicherbedarfs von Mergesort, ob sie direkt mit der Datenbank arbeiten oder besser mit einem "Adressarray". Damit ist gemeint, dass man ein Array mit Indizes (bzw Adressen) auf die Elemente der Datenbank nach einem selbstgewählten Vergleichskriterium sortiert. Dafür benötigen Sie eine Funktion `bool kleinerodergleich(Student* Dbase, int ind1, int ind2)`, die zurückgibt, ob beispielsweise `Dbase[ind1].matnr <= Dbase[ind2].matnr` ist.

(15 Punkte)

Die Programmieraufgabe wird in der Woche vom 04.12.2017 bepunktet

Programmieraufgabe 2. (Die Türme von Hanoi)

Implementieren Sie die "Türme von Hanoi" in C/C++. Nutzen Sie geeignete Arrays für die aktuelle Belegung der Stäbe (oder kodieren Sie diese als Binärzahl). Geben Sie nach jeder Aktion die aktuelle Belegung aus, z.B. zeilenweise für jeden Stab, wobei jede Scheibe durch eine Nummer kodiert ist.

(5 Zusatz- Punkte)

Die Programmieraufgabe wird in der Woche vom 04.12.2017 bepunktet