



Algorithmische Mathematik I

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



Übungsblatt 7.

Abgabe am **27.11.2017** vor der Vorlesung.

Aufgabe 1. (Bubblesort)

- a. Gegeben sei folgende Zahlenfolge:

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

Sortieren Sie diese Zahlenfolge mit dem Bubblesort-Algorithmus! Geben Sie dabei in jedem Schleifendurchlauf die bis dahin entstandene Permutation an, etwa so:

Anfang: 7, 17, 5, 3, 2, 11, 13

$m = 6, i = 1$: \vdots

$m = 6, i = 2$: 7, 5, 17, 3, 2, 11, 13

...

- b. Jemand sortiert eine Permutation der Zahlen $1, \dots, 8$ mit Bubblesort. Gerade als die 8 ihre Position ganz hinten erreicht hat, verschüttet er Kaffee auf seine Lösung. Nun ist da nur noch folgendes zu lesen (ein * steht dabei für eine unleserliche Zahl):

4, 3, *, 8, *, *, *, *

(Einige ganz unleserliche Zeilen)

*, *, *, *, *, *, *, 6

4, 3, 2, *, *, *, *, 8

Erklären Sie, warum er sich entspannen kann, da seine Lösung ohnehin falsch ist!

- c. Nehmen Sie an, eine Implementierung des Bubblesort benötigt zum Sortieren von 1000 Adressen im Durchschnitt eine Hundertstelstesekunde. Wie lange würde es in etwa dauern die Adressen der 81 Millionen Einwohner Deutschlands zu sortieren? Wie lange für die aller 7 Milliarden Menschen?

(4 + 4 + 2 = 10 Punkte)

Aufgabe 2. (Fibonacci-Folge)

Gegeben sei die Fibonacci-Folge

$$f_n = f_{n-1} + f_{n-2}, \quad n > 2, \quad f_1 = f_2 = 1.$$

- a. Zeigen Sie, dass die Folgenglieder der expliziten Formel

$$f_n = \frac{a^n - b^n}{\sqrt{5}} \quad \text{mit} \quad a = \frac{1 + \sqrt{5}}{2} \quad \text{und} \quad b = \frac{1 - \sqrt{5}}{2}$$

genügen.

b. Zeigen Sie, dass $\lim_{n \rightarrow \infty} f_{n+1}/f_n$ existiert. Berechnen Sie diesen Grenzwert.

Hinweis. Zeigen sie zunächst, dass $a = 1 + \frac{1}{a}$ und $b = 1 + \frac{1}{b}$ gilt.

(5 Punkte)

Aufgabe 3. (Zeiger in C/C++)

Nach den Definitionen

```
int i=5;
int *pi, pj;
char *pc, pd;
werden folgende Zuweisungen (einzeln) ausgeführt
pi=i;
pi=&i;
*pi=i;
*pi=&i;
pi=pj;
pc=&pd;
pi=pc;
pd=*pi;
*pi=i**pc;
pi=0;
```

Entscheiden Sie, welche Zuweisungen syntaktisch korrekt sind und geben sie nach jeder syntaktisch korrekten Zuweisung den Wert der linken Seite an (bei Zeigern dereferenzieren).

(5 Punkte)

Programmieraufgabe 1. (Fibonacci)

- Implementieren Sie eine rekursive C/C++ Funktion, die fuer ein Argument n die n -te Fibonaccizahl ausgibt.
- Implementieren Sie eine Funktion, die für ein Argument n die ersten n Fibonaccizahlen in einem array speichert und anschließend ausgibt. Dafür soll das array als Parameter übergeben werden. Implementieren Sie zwei Varianten. Die eine Implementierung soll Teilaufgabe a) benutzen und die andere möglichst effizient vorgehen.

Die Programmieraufgabe ist eine Präsenzaufgabe und wird nicht bepunktet.

Programmieraufgabe 2. (Addition mit Listen)

Gegeben sei folgendes Beispielprogramm:

```
1 #include <limits.h>
  #include <iostream>
3
  using namespace std;
5
  int main()
7  {
    unsigned int i = UINT_MAX;
9    unsigned int j = UINT_MAX;
    cout << i+j;
11 }
```

Kompilieren Sie dieses Programm und schauen Sie sich die Ausgabe an. Was bemerken Sie?

In manchen Fällen wollen Sie auch ganze Zahlen, die über die maximalen Werte der Standarddatentypen gehen, verwenden. Hierfür eignen sich dynamische Datenstrukturen, wie einfach verkettete Listen. In diesen Listen können sehr große Ganzzahlen in folgender Form gespeichert werden:

$$127_{10} = 1 \cdot 10^2 + 2 \cdot 10^1 + 7 \cdot 10^0 \Rightarrow \text{Liste } \boxed{7} \rightarrow \boxed{2} \rightarrow \boxed{1}$$

Mit dieser Darstellung der Ganzzahl können nun beliebig große Zahlen addiert werden:

$$\begin{array}{r} \boxed{7} \rightarrow \boxed{2} \rightarrow \boxed{1} + \\ \boxed{7} \rightarrow \boxed{0} \rightarrow \boxed{0} \end{array}$$

Nun kann man, wie gewohnt aus Grundschulzeiten, von hinten die Zahlen in der Liste addieren und den Übertrag zur nächsten Stelle weiter reichen.

$$\begin{array}{l} \text{Erste Stelle: } 7 + 7 = 14 \text{ Übertrag: } 1 \\ \text{Zweite Stelle: } 2 + 0 + 1 = 3 \text{ Übertrag: } 0 \\ \text{Dritte Stelle: } 1 + 0 + 0 = 1 \text{ Übertrag: } 0 \end{array}$$

In der resultierenden Liste steht nun das Ergebnis $134 = \boxed{4} \rightarrow \boxed{3} \rightarrow \boxed{1}$.

- a. Definieren Sie den Datentyp `struct element`, der eine Ziffer und einen Zeiger auf das nächste Element enthält (einfach verkettete Liste).
- b. Addieren Sie folgenden Zahlen mit ihrem neuen Datentyp:
 - (a) $65535 + 65535$
 - (b) $18446744073709551615 + 9223372036854775807$
- c. Wie könnte man eine Langzahlmultiplikation implementieren?

Hinweise:

- a. Die aktuelle Ziffer bekommen Sie mittels $(ziffer1 + ziffer2 + ?) \% 10$.
- b. Den Übertrag bekommen Sie mittels $(int)((ziffer1 + ziffer2 + ?) / 10)$.

Die Programmieraufgabe ist eine Präsenzaufgabe und wird nicht bepunktet.