Algorithmische Mathematik

Wintersemester 2013
Prof. Dr. Marc Alexander Schweitzer und
Dr. Einar Smith
Patrick Diehl und Daniel Wissel

universitätbonn

Übungsblatt 1.

Abgabe am **28.10.2013 zwischen Vorlesung A und Vorlesung B**.

Aufgabe 1. (Bubblesort verstehen)

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:

$$\begin{aligned} \text{Anfang:} \quad & 7, 17, 5, 3, 2, 11, 13 \\ m = 6, i = 1: \quad & \vdots \\ m = 6, i = 2: \quad & 7, 5, 17, 3, 2, 11, 13 \\ \dots & \end{aligned}$$

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):

$$\begin{gathered} 4, 3, *, 8, *, *, *, * \\ \text{(Einige ganz unleserliche Zeilen)} \\ *, *, *, *, *, *, *, 6 \\ 4, 3, 2, *, *, *, *, 8 \end{gathered}$$

Trösten Sie ihn, indem Sie ihm erklären, warum seine Lösung nicht richtig gewesen sein kann!

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. (Laufzeitanalyse)

a. Zeigen Sie, dass die best-case Laufzeit für Bubblesort in $\mathcal{O}(n)$ liegt.

b. Zeigen Sie, dass die worst-case Laufzeit für Bubblesort in $\mathcal{O}(n^2)$ liegt.

Welche Anforderung wird an die Eingabe gestellt, um die best-case Laufzeit oder worst-case Laufzeit zu erzielen?

(3 + 3 = 6 Punkte)

Aufgabe 3. (Mergesort verstehen)

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 $\boxed{1, 2, 3}$ $\boxed{4, 5, 6}$.

(4 Punkte)

Programmieraufgabe 1. (Polynomauswertung)

Ein Polynom n,ten Grades ist definiert durch

$$p(x) = \sum_{i=0}^{n} a_i x^i$$

mit $a_i \in \mathbb{R},\ i = 0, \ldots, n$ und $x \in \mathbb{R}$. Implementieren Sie eine Funktion *polynom(a,n,x)*, die das Polynom n,ten Grades an der Stelle x auswertet. Die Koeffizienten a_i werden in einem *double*,Array übergeben.
Testen Sie Ihr Programm für die Auswertung des Polynoms

$$p(x) = 3x^5 - 4x^4 + 3x^2 - x + 0.079, \quad x = 0.125.$$

(4 Punkte)

Abgabe am 28.10.2013 zwischen der Vorlesung A und Vorlesung B

Programmieraufgabe 2. (Maximum von Array-Elementen)

Schreiben Sie den in der Vorlesung präsentierten Code zur Bestimmung des Maximums von *int*-Array-Elementen derart um, dass sie eine Funktion *int intmax(int[] arr, int n)* erhalten, welche als Rückgabewert das maximale Element der n Array-Elemente liefert. Testen Sie die Funktion anhand von drei selbst gewählten Beispielen.

(4 Punkte)

Abgabe am 28.10.2013 zwischen der Vorlesung A und Vorlesung B

Programmieraufgabe 3. (Ackermannfunktion)

Die Ackermannfunktion ist definiert durch

$$\begin{aligned} a(0,y) &= y+1, \\ a(x,0) &= a(x-1,1), \quad x>0, \\ a(x,y) &= a(x-1,a(x,y-1)) \quad x,y>0 \end{aligned}$$

Implementieren Sie ein C-Programm, dass die Ackermannfunktion rekursiv berechnet. Experimentieren Sie mit ihren beiden Funktionen und betrachten was schon bei kleinen Werten für x und y passiert.

(8 Punkte)

Abgabe am 28.10.2013 zwischen der Vorlesung A und Vorlesung B

Programmieraufgabe 4. (Insertionsort)

Die intuitivste Variante eine Menge Zahlen zu sortieren, ist vergleichbar der Sortierung eines Skatblatts. Dort werden die Karten verdeckt auf einen Stapel gelegt, einzeln aufgedeckt und dann einsortiert. Der Insertionsort verwendet dieses Vorgehen. Implementieren Sie den Insertionsort in C und testen den Algorithmus auf einer beliebigen Eingabe.

(8 Punkte)

Abgabe am 28.10.2013 zwischen der Vorlesung A und Vorlesung B