

Übungsaufgaben zum Vorkurs Informatik Wintersemester 2017/2018

Teil I

Aufgabe 1

Geben Sie sieben Teilgebiete der Informatik an.

Aufgabe 2

Nennen Sie wesentliche Hardware-Komponenten eines PCs und, falls sinnvoll, ihre typischen Leistungskennzahlen.

Aufgabe 3 [Für die erste Übung am Rechner]

Lernen Sie die Java-Programmierungsumgebung kennen, indem Sie das Programm „HelloWorld“ aus der Vorlesung eingeben, übersetzen und ausführen lassen.

Teil II

Aufgabe 1

Geben Sie eine Definition des Begriffs „Algorithmus“. Was sind die vier grundlegenden Eigenschaften?

Aufgabe 2 [Für die zweite Übung am Rechner]

a)

Schreiben Sie einen Algorithmus $\max(a_0, a_1, \dots, a_n)$, der den größten Wert der Zahlenmenge $\{a_0, a_1, \dots, a_n\}$ findet.

b)

Schreiben Sie ein Java-Programm `ProgramMaxSuche` analog zum Programm `ProgrammMinSuche` aus der Vorlesung. Verwenden Sie zum Testen auch einmal eine andere Zahlenmenge.

Aufgabe 3 [Für die zweite Übung am Rechner]

a)

Schreiben Sie einen Algorithmus *fuenfteilbar*(a_0, a_1, \dots, a_n) in Pseudocode, der „true“ zurückliefert, wenn die Zahlenmenge $\{ a_0, a_1, \dots, a_n \}$ eine Zahl enthält, die durch 5 teilbar ist, und sonst „false“ zurückgibt. Die Teilbarkeit kann mit „modulo“ getestet werden: a modulo b liefert den Rest bei der ganzzahligen Teilung von a durch b .

b)

Schreiben Sie eine Java-Funktion `static boolean fuenfteilbar(int[] a)`, die dies leistet. Java repräsentiert modulo durch `%`, d.h. `a % b` steht für $a \bmod b$.

c)

Schreiben Sie ein Testprogramm, dessen Hauptprogramm ein Array deklariert und initialisiert, dann die Methode `fuenfteilbar` aufruft und das Ergebnis des Aufrufs auf dem Bildschirm ausgibt.

Aufgabe 4 [Für die dritte Übung am Rechner]

a)

Schreiben Sie einen Algorithmus *Invers*($\text{int}[] a$) in Pseudocode, der die Reihenfolge der Werte im Array a umdreht. Z.B. soll für den Fall, dass $a[0]=4$, $a[1]=7$ und $a[2]=9$ ist, $a[0]=9$, $a[1]=7$ und $a[2]=4$ herauskommen. Bei dem Algorithmus soll kein zusätzliches Hilfs-Array verwendet werden (sog. in-place Operation).

b)

Implementieren Sie den gefundenen Algorithmus als Java-Funktion `static void invers(int[] a)`. Diese Funktion soll zunächst das Original-Array und dann das inverse Array auf dem Bildschirm ausgeben (`System.out.println`).

c)

Schreiben Sie ein Testprogramm, dessen Hauptprogramm ein Array deklariert und initialisiert und dann die Methode `invers` aufruft.

Aufgabe 5 [Für die dritte Übung am Rechner]

a)

Schreiben Sie einen Algorithmus *mehrfach*(a_0, a_1, \dots, a_n) in Pseudocode, der in einer Menge $\{ a_0, a_1, \dots, a_n \}$ von $n+1$ Zahlen herausfindet, ob zwei Zahlen den gleichen Wert haben. Der Algorithmus soll „true“ zurückgeben, wenn dies der Fall ist, sonst „false“.

b)

Schreiben Sie eine Java-Funktion `static boolean mehrfach(int[] a)`, die dies leistet.

c)
Schreiben Sie ein Testprogramm, dessen Hauptprogramm ein Array deklariert und initialisiert, dann die Methode `mehrfach` aufruft und das Ergebnis des Aufrufs ausdrückt.

d)
Schätzen Sie den maximalen Zeitaufwand Ihres Algorithmus in Abhängigkeit von n ab.

Aufgabe 6

Die Primzahlen in einem Intervall $I(n)=\{2, \dots, n\}$ ganzer Zahlen lassen sich dadurch berechnen, dass zunächst alle Zahlen aus $I(n)$ entfernt werden, die durch 2 teilbar sind und größer als 2 sind. Aus der Menge der verbleibenden Zahlen werden dann alle diejenigen Zahlen entfernt, die durch 3 teilbar sind und größer als 3 sind, usw., bis $n/2$ erreicht ist.

a)
Schreiben Sie einen Algorithmus in Pseudocode, der dies leistet.

b)
Schätzen Sie den maximalen Zeitaufwand Ihres Algorithmus in Abhängigkeit von n ab.

Aufgabe 7 [Für die vierte Übung am Rechner]

a)
Die Summe der ganzen Zahlen zwischen 1 und n soll berechnet werden. Schreiben Sie einen Algorithmus `summeBis(n)` in Pseudocode, der dies leistet.

b)
Die Summe s_n der ganzen Zahlen zwischen 1 und n kann rekursiv wie folgt definiert werden:

$$s_1 := 1, s_n := s_{n-1} + n \text{ für } n > 1.$$

Schreiben Sie einen Algorithmus `rekSummeBis(n)` in Pseudocode, der die Berechnung auf diese Weise ausführt.

c)
Schreiben Sie Java-Funktionen `static int summeBis(int n)` und `static int rekSummeBis(int n)`, die diese Algorithmen implementieren.

d)
Schreiben Sie ein Testprogramm, das beide Funktionen mit dem gleichen Wert aufruft und jeweils das Ergebnis auf dem Bildschirm ausgibt.

Aufgabe 8 [Für die vierte Übung am Rechner]

Der größte gemeinsame Teiler zweier positiver ganzer Zahlen a , b lässt sich mit folgendem rekursiv definierten Verfahren berechnen:

$$\begin{aligned} \text{ggt}(a, 1) &= 1, \\ \text{ggt}(1, b) &= 1, \\ \text{ggt}(a, a) &= a, \\ \text{ggt}(a, b) &= \text{ggt}(a-b, b) \text{ für } a > b > 1, \\ \text{ggt}(a, b) &= \text{ggt}(a, b-a) \text{ für } b > a > 1. \end{aligned}$$

Schreiben Sie eine Java-Funktion `static int ggt(int a, int b)`, die den größten gemeinsamen Teiler so berechnet. Schreiben Sie ein Testprogramm, das diese Funktion mit aktuellen Parametern aufruft und die Parameter sowie das Ergebnis auf dem Bildschirm ausgibt.

Teil III

Aufgabe 1

a)

Welche der folgenden Angaben ist richtig:

$$(1) 4n^2 + 5n - 1 = O(n^2) \quad (2) 4n^2 + 5n - 1 = O(n) \quad (3) 4n^2 + 5n - 1 = O(n^3)$$

b)

Geben Sie eine möglichst kleine Ordnung an, die das asymptotische Wachstum der folgenden Funktionen beschreibt:

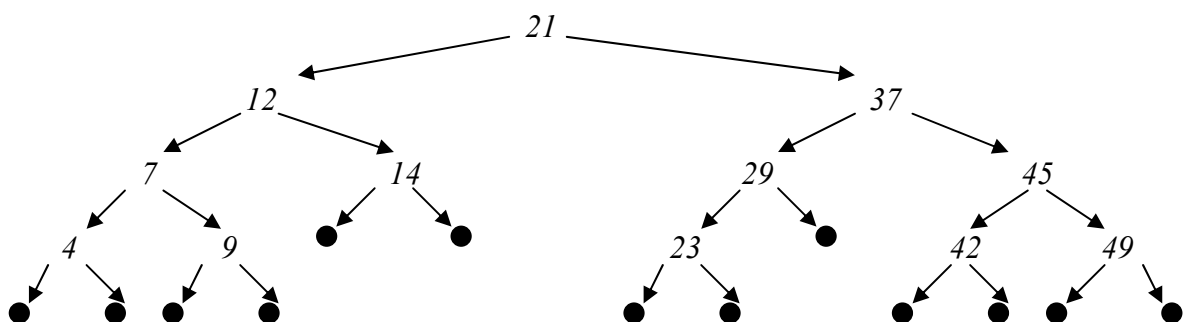
$$(1) f(n) = 5n + 17 \quad (2) g(n) = 3 + 5n \log n \quad (3) h(n) = 2n^3 - 19n^2 + 17n - 2$$

c)

Führen Sie das Verfahren der binären Suche für die Menge $S = \{47, 21, 27, 6, 18, 28\}$ und die Suchwerte 27 und 17 durch. Geben Sie dabei die Intervalle an, die nacheinander durchlaufen werden.

Aufgabe 2

Gegeben sei der folgende binäre Suchbaum



a)
Ist der Baum ausgeglichen im Sinne der Definition der Vorlesung?

b)
Geben Sie die Knoten an, die bei der Suche nach 15, 45 und 20 durchlaufen werden.

Aufgabe 3

Gegeben ist die Zahlenmenge $S = \{87, 23, 27, 16, 18, 28\}$.

a)
Zeichnen Sie einen ausgeglichenen binären Suchbaum für S .

b)
Zeichnen Sie die Belegung der Hash-Datenstruktur für S , die ein Hash-Array mit $n=11$ Elemente hat und deren Hash-Funktion durch $i(s)=s \bmod (n-1)$ definiert ist.

c)
Geben Sie ein n an, so dass für die oben gegebene Menge S keine Kollision in der Hash-Datenstruktur auftritt, d.h. eine Zelle des Hash-Arrays nur auf höchstens einen Eintrag verweist.

Aufgabe 4

Beweisen Sie durch vollständige Induktion, dass die Summe

$$s(n) = 1^3 + 2^3 + 3^3 + \dots + n^3$$

der mit 3 potenzierten ganzen Zahlen von 1 bis n gleich $(n(n+1)/2)^2$ ist.

Aufgabe 5

Beweisen Sie durch vollständige Induktion, dass die Anzahl $k(h)$ an Knoten, die ein binärer Baum der Höhe h haben kann, höchstens 2^h-1 ist, d.h. $k(h) \leq 2^h-1$.

Aufgabe 6

Angenommen, es ist von einem Algorithmus bekannt, dass sein maximaler Zeitaufwand $T(n)$ folgender Beziehung genügt:

$$T(n) = T(n-1) + O(1) \text{ für } n > 1.$$

Geben Sie eine möglichst kleine gültige Wachstumsordnung (O-Notation) für $T(n)$ an. Beweisen Sie, dass $T(n)$ diese Beziehung erfüllt.

Teil IV

Aufgabe 1

Was sind die Basiskomponenten eines Rechners in der von-Neumann-Architektur?

Aufgabe 2

Beschreiben Sie den Ablauf der Befehlsausführung bei einem von-Neumann-Rechner.

Aufgabe 3

Die Funktionstabelle der Booleschen Funktion *xor* (entweder oder) sieht so aus:

<i>a</i>	<i>b</i>	<i>xor(a,b)</i>
0	0	0
0	1	1
1	0	1
1	1	0

a)

Geben Sie eine Boolesche Formel mit *and*, *or* und *not*-Operationen für *xor* an.

b)

Zeichnen Sie einen Schaltkreis, der die Boolesche Formel realisiert.

Aufgabe 4

Stellen Sie die Funktionstabelle der Booleschen Funktion $f = a \bar{b} c + \bar{a} b \bar{c}$ auf.

Aufgabe 5

Was macht ein Compiler, was ein Interpreter?

Aufgabe 6

a)

Von welchem Typ ist die Grammatik

$$G=(T,N,P,S) \text{ mit } T=\{a,b\}, N=\{S\}, P=\{S \rightarrow ab, S \rightarrow aSb\}?$$

b)

Geben Sie die Sprache $L(G)$ an, die von G erzeugt wird.

Aufgabe 7

Geben Sie eine reguläre Grammatik G an, deren Alphabet T nur das Zeichen c enthält und deren Sprache $L(G)$ alle Worte umfasst, in der das Zeichen c mit gerader Anzahl auftritt? Falls Sie keine reguläre Grammatik finden, genügt auch eine kontextfreie.