

Aufgabe 5.1

Bei der Angabe der Laufzeit soll in dieser Aufgabe die Anzahl der im worst case anfallenden elementaren Anweisungen möglichst genau bestimmt werden (keine O -Notation verwenden).

- (a) Geben Sie für die folgende Methode `DoSomething1` kurz an, was für das übergebene Array `a` berechnet wird. Geben Sie außerdem die Laufzeit der Methode im worst case als Funktion der Länge n des Arrays `a` an.

```
static int DoSomething1(int[] a)
{
    int s=0;
    for (int i=0;i<a.length;i=i+2) {
        s=s+a[i];
    }
    return s;
}
```

(2 Punkte)

- (b) Geben Sie für die folgende Methode `DoSomething2` auch kurz an, was für die übergebene ganze Zahl `n` berechnet wird. Geben Sie außerdem die Laufzeit der Methode im worst case als Funktion von `n` an. Es genügt, wenn Sie denn Fall $n \geq 2$ betrachten.

```
static int DoSomething2(int n)
{
    int x=2;
    while (x<n) {
        x=2*x;
    }
    return x;
}
```

(2 Punkte)

Aufgabe 5.2

- (a) Geben Sie für die folgenden Paare von Funktionen f und g jeweils eine geeignete Konstante $c > 0$ und eine geeignete natürliche Zahl $n_0 \in \mathbb{N}$ an, die belegen, dass $f \in O(g)$ gilt.

(i) $f(n) = n^3 + 12 \log_2 n$, $g(n) = n^3$

(ii) $f(n) = 4n^2 + n + 10$, $g(n) = n^4$

(iii) $f(n) = \log_2 n$, $g(n) = \log_{10} n$

(6 Punkte)

- (b) Versuchen Sie zu argumentieren, dass für alle Funktionen

$$f : \mathbb{N} \rightarrow \mathbb{R}, \quad g : \mathbb{N} \rightarrow \mathbb{R} \text{ und } h : \mathbb{N} \rightarrow \mathbb{R}$$

gilt: Aus $f \in O(g)$ und $g \in O(h)$ folgt $f \in O(h)$.

(4 Punkte)

Zusatzaufgabe 5.3

Was ist die Laufzeit im worst case der Methode

```
static int findMin(int[] listOfNumbers)
```

im Programm `Beispiel03.java` aus der Vorlesung als Funktion der Anzahl n von Elementen des Arrays `listOfNumbers`.

(3+Punkte)