11. Übungsblatt - Informatik 1 - Lösungsbeispiele

Aufgabe 1 (Zeitaufwand)

Bestimmen Sie den Zeitaufwand zur Berechnung des größten gemeinsamen Teilers zweier Zahlen a und b für den schlimmsten und besten Fall. Dabei sei die Eingabekomplexität $n := max\{a, b\}$.

```
public int ggT(int a, int b) {
  while (a != b) {
    if (a > b) {
      a = a - b;
    } else {
      b = b - a;
    }
  }
}
```

Lösungsvorschlag:
Der beste Fall ist $n = a = b$. Die Schleife bricht sofort ab. Der Zeitaufwand ist $T_{bc}(n) = O(1)$.

Der schlimmste Fall ist $n = a > b = 1$. Die Schleife wird $n-1$ mal durchlaufen. Der Zeitaufwand ist $T_{wc}(n) = O(n)$.

Aufgabe 2 (Zeitaufwand)

Bestimmen Sie möglichst genau den Zeitaufwand in Abhängigkeit von n des folgenden Programms und geben Sie ihn im O-Kalkül an.

```
int x = 1;
int y = 0;
for (int i = 0; i < n; i++) {
  for (int j = 0; j < n; j += x) {
    y++;
  }
  x = x * 3;
}
```

Lösungsvorschlag:
Es reicht die Anweisung y++ zu zählen. Die Schrittweite der inneren Schleife startet bei $x = 1$ und erhöht sich nach jedem Durchlauf der äußeren Schleife um 3. Bei erstem mal wird sie also n-mal durchlaufen, dann $\frac{n}{3}$, dann $\frac{n}{3^2}$ und so weiter. Der Gesamtzweitaufwand ist also höchstens

$$\begin{aligned} T(n) &= n \cdot (\textstyle\sum_{i=0}^{n-1} \frac{1}{3^i}) \\ &\leq n(\frac{1}{1-\frac{1}{3}}) \\ &= n(\frac{3}{2}) \\ &= O(n) \end{aligned}$$

Hinweis: Aus der Vorlesung ist bekannt, dass für alle $0 \leq q < 1$ gilt: $1 + q + \ldots + q^{n-1} = \frac{1-q^n}{1-q} < \frac{1}{1-q}$

Aufgabe 3 (Rekurrenzgleichung)

Lösen Sie folgende Rekurrenzgleichung. Beweisen Sie Ihre Behauptung mit vollständiger Induktion. Geben Sie das Ergebnis vereinfacht im O-Kalkül an.

$$T(n) = 2T(\frac{n}{2}) + cn^2, n > 1$$

Lösungsvorschlag:
Um eine Lösung zu erraten, wenden wir die Rekurrenz mehrfach auf sich an:

$$\begin{aligned}
T(n) &= 2T(\tfrac{n}{2}) + cn^2 \\
&= 2\left(2T(\tfrac{n}{2^2}) + c(\tfrac{n}{2})^2\right) + cn^2 \\
&= 2^2T(\tfrac{n}{2^2}) + c\tfrac{n^2}{2} + cn^2 \\
&= 2^2\left(2T(\tfrac{n}{2^3}) + c(\tfrac{n}{2^2})^2\right) + c\tfrac{n^2}{2} + cn^2 \\
&= 2^3T(\tfrac{n}{2^3}) + c\tfrac{n^2}{2^2} + c\tfrac{n^2}{2} + cn^2 \\
&= \ldots, \text{für } = log_2(n) \\
&= 2^kT(1) + c\tfrac{n^2}{2^{k-1}} + \ldots + c\tfrac{n^2}{2^2} + c\tfrac{n^2}{2} + cn^2 \\
&= 2^kc + cn^2(\tfrac{1}{2^{k-1}} + \ldots + c\tfrac{1}{2^2} + c\tfrac{1}{2} + 1) \\
&\leq nc + cn^2(\tfrac{1}{1-\frac{1}{2}}) \text{ (siehe Hinweis Lösung vorherige Aufgabe)} \\
&= nc + 2 \cdot cn^2
\end{aligned}$$

Induktionsbehauptung: $T(n) \leq nc + 2 \cdot cn^2$ für alle $n \geq 1$.
Induktionsanfang für $n = 1$: $T(1) = c < 1 \cdot c + 2 \cdot c \cdot 1^2$
Induktionsvoraussetzung (IV): Induktionsbehauptung gilt für alle $n \geq 1$.
Induktionsschluss von n auf $2 \cdot n$:

$$\begin{aligned}
T(2n) &= 2T(n) + c \cdot n^2 \\
&\leq 2(nc + 2 \cdot cn^2) + c \cdot n^2, \text{ IV für } n \\
&= 2nc + 4 \cdot cn^2 + c \cdot n^2 \\
&\leq 2nc + 8 \cdot cn^2 \\
&= (2n)c + 2 \cdot c(2n)^2
\end{aligned}$$

Es gilt also $T(n) = O(n^2)$