Prof. Dr. Leif Kobbelt
Stefan Dollase, Ira Fesefeldt, Alexandra Heuschling, Gregor Kobsik

Übung 3

Tutoraufgabe 1 (Laufzeitanalyse):

Gegeben sei ein Algorithmus, der für ein Array von Integern überprüft, ob die Einträge aufsteigend sortiert sind:

```
1 def is_sorted(a):
2     i = 1
3     while i < len(a):
4         if a[i - 1] > a[i]:
5             return False
6         i += 1
7     return True
```

Bei Betrachtung der Laufzeit wird angenommen, dass die Vergleiche jeweils eine Zeiteinheit benötigen (siehe Zeilen 3 und 4). Die Laufzeit aller weiteren Operationen wird vernachlässigt.
Sei $n \in \mathbb{N}$ die Länge des Arrays a.

a) Bestimmen Sie in Abhängigkeit von n die Best-case Laufzeit $B(n)$. Begründen Sie Ihre Antwort. Geben Sie für jedes n ein Beispielarray an, bei dem diese Laufzeit erreicht wird.

b) Bestimmen Sie in Abhängigkeit von n die Worst-case Laufzeit $W(n)$. Begründen Sie Ihre Antwort. Geben Sie für jedes n ein Beispielarray an, bei dem diese Laufzeit erreicht wird.

c) Bestimmen Sie in Abhängigkeit von n die Average-case Laufzeit $A(n)$. Begründen Sie Ihre Antwort. Hierzu nehmen wir für $n \geq 2$ folgende Verteilung der Eingaben an:

- $\Pr(\mathtt{a}[0] = 1) \;=\; 1$,
- $\Pr(\mathtt{a}[\mathtt{n} - 1] = 0) \;=\; 1$ und
- für alle $\mathtt{i} \in \{1, \ldots, n-2\}$ gilt: $\Pr(\mathtt{a}[\mathtt{i}] = 0) \;=\; \Pr(\mathtt{a}[\mathtt{i}] = 1) \;=\; 0.5$.

Der erste Eintrag ist also immer 1 und der letzte Eintrag ist immer 0. Die übrigen Einträge sind mit gleicher Wahrscheinlichkeit entweder 0 oder 1. Insbesondere ist die Wahrscheinlichkeit, dass das Array aufsteigend sortiert ist immer 0. Für $n < 2$ ist die Verteilung der Eingaben nicht relevant.

Hinweise:

- Wir gehen davon aus, dass die Indizierung von Arrays mit 0 beginnt.
- Geben Sie die geforderten Laufzeiten in geschlossener Form (d.h. ohne Summenzeichen, $\Pr(\cdots)$ oder ähnliches) an.

Tutoraufgabe 2 (Rekursionsgleichung):

Finden Sie für die Rekursionsgleichung F, die wir als

$$F(1) = 1 \text{ und } F(n) = F(n-1) + n^2 + n \text{ für } n > 1$$

definieren, ein Polynom $g : \mathbf{R}_+ \to \mathbf{R}_+$ als obere Abschätzung an, welche nicht rekursiv definiert ist. Beweisen Sie anschließend per vollständiger Induktion, dass ihr Vorschlag tatsächlich eine obere Abschätzung ist, also dass $F(n) \leq g(n)$ ist. Geben Sie schließlich auch eine Abschätzung der Funktion g in O-Notation an.

Tutoraufgabe 3 (Master Theorem):

Benutzen Sie das Master Theorem um für folgende Rekursionsgleichungen $T(n)$ eine möglichst gute O-Klasse anzugeben, in der $T(n)$ liegt.

a)

$$T(1) = 1$$
$$T(n) = 16 \cdot T(\frac{n}{2}) + n^5 + \log_4(n) \qquad \text{for } n > 1$$

b)

$$T(1) = 2$$
$$T(n) = 27 \cdot T(\frac{n}{3}) + n^3 + n^2 \qquad \text{for } n > 1$$

c)

$$T(1) = 3$$
$$T(n) = 26 \cdot T(\frac{n}{5}) + n \cdot \log_2 n \qquad \text{for } n > 1$$