Fachbereich Mathematik
PD Dr. Ulf Lorenz
M.Sc. Franziska Kartzow

TECHNISCHE UNIVERSITÄT DARMSTADT

SS 2010
6. Mai 2010

2. Übungsblatt zur „Algorithmischen Diskreten Mathematik“

Gruppenübung

Aufgabe G1 (Master Theorem)

Bestimmen Sie - sofern möglich - feste Schranken für die folgenden Rekurrenzen:

- $T(n) = 4T(\frac{n}{2}) + n^3$
- $T(n) = 4T(\frac{n}{2}) + n$
- $T(n) = 4T(\frac{n}{2}) + n^2$
- $T(n) = 4T(\frac{n}{2}) + n^2 \log n$.

Aufgabe G2 (Komplexität)

(a) Seien $f, t : \mathbb{N} \to \mathbb{R}$ gegeben mit $f \in \mathcal{O}(t)$. Zeigen Sie, dass $\mathcal{O}(f) + \mathcal{O}(t) \subseteq \mathcal{O}(t)$ und $\mathcal{O}(f) + \mathcal{O}(f) \subseteq \mathcal{O}(t)$.

(b) Gilt $3^{3+n} \in \mathcal{O}(3^n)$?

(c) Gilt $3^{3n} \in \mathcal{O}(3^n)$?

(d) Zeigen Sie $\mathcal{O}(f) \cdot \mathcal{O}(g) = \mathcal{O}(f \cdot g)$

Hinweis: Bei reellen Funktionen ersetzt man in der Definition von $\mathcal{O}(f)$ bzw. $\mathcal{O}(g)$ $f(n)$ bzw $g(n)$ mit $|f(n)|$ bzw. $|g(n)|$.

Aufgabe G3

(a) Wir haben zwei Algorithmen gegeben:

- Algorithmus A mit einer Laufzeit von $\mathcal{O}(f)$
- Algorithmus B mit einer Laufzeit von $\mathcal{O}(g)$

Wir betrachten zwei weitere Algorithmen, 1 und 2.

Algorithm 1

INPUT : $n \in \mathbb{N}$
for $i = 1, ..., 100$ **do**
  Führe Algorithmus A aus
end for
for i= $1, ..., \frac{n}{2}$ **do**
  Führe Algorithmus B aus
end for

Algorithm 2

```
if  n ≥ 30 then
   Führe Algorithmus A aus
else
   Führe Algorithmus B aus
end if
```

Wir wissen, dass $f \in \Omega(g)$. Ermitteln Sie eine möglichst gute Laufzeitabschäztung für die beiden Algorithmen.

(b) Betrachten Sie Algorithmus 3 und geben Sie eine möglichst gute Abschätzung für seine Laufzeit an. Begründen Sie Ihre Aussage.

Algorithm 3

```
INPUT : n ∈ ℕ
for i =1,..., n do
   for j = 1,...,n/2 do
      a=3 · b
      c = a +b
   end for
   n = 1/2 · n
end for
```

Aufgabe G4

Sortieren Sie die Funktionen

$$n^2,\ \sqrt{n},\ n!,\ n^n,\ n$$

nach aufsteigender Komplexität unter Verwendung der o-Notation. Bestimmen Sie jeweils n_0.

Zur Erinnerung :

$$f \in o(g)\ :\ \Leftrightarrow\ \forall c > 0\ \ \exists n_0 \in \mathbb{N}\ \forall n \geq n_0 : 0 \leq f(n) < c \cdot g(n)$$

Hausübung

Aufgabe H1 (Inklusion-Exklusionsprinzip) (6 Punkte)

(a) Für eine Menge X bezeichnen wir mit

$$\binom{X}{k}$$

die Menge aller k-elementigen Teilmengen von X. Es seien $A_1, ..., A_n$ endliche Mengen. Zeigen Sie, dass

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{k=1}^{n} (-1)^{k-1} \sum_{I \in \binom{\{1,2,..,n\}}{k}} \left|\bigcap_{i \in I} A_i\right|$$

gilt.

(b) In einer reinen Jungen-Klasse, spielen 18 Jungen gerne Schach, 23 bevorzugen Fussball, 21 Rad fahren und 17 Wandern. Sowohl Schach als auch Fussball spielen 9 der Jungen. Wir wissen außerdem, dass 7 Jungen Schach und Rad fahren mögen, 6 begeistern sich für Schach und Wandern, 12 mögen Fussball und Rad fahren, während 9 Jungen für Fussball und Wandern und 12 für Rad fahren und Wandern sind. Es gibt 4 Jungen, die Schach, Fussball und Rad fahren mögen, 3 Jungen, die Schach, Fussball und Wandern bevorzugen, 5 Jungen, die Schach, Rad fahren und Wandern mögen, während 7 gerne Fussball spielen, Rad fahren und wandern. Schließlich gibt es noch 3 Jungen, die sich allen vier Aktivitäten gerne widmen. Wir wissen außerdem, dass jeder in der Klasse mindestens eine dieser Aktivitäten mag.Wie viele Jungen gibt es in der Klasse?

Aufgabe H2 (Asymptotik) (4 Punkte)

Zeigen Sie, dass für alle $r_1, r_2 \in \mathbb{R}^+$ gilt $n^{r_1} \in \mathcal{O}(n^{r_2})$ und ${r_1}^n \in \mathcal{O}({r_2}^n)$ genau dann wenn $r_1 \leq r_2$.

Aufgabe H3 (Sortieralgorithmus) (4 Punkte)

Der Algorithmus *SortList* sortiert eine Folge von Zahlen nach aufsteigender Reihenfolge.

Algorithm 4 SortList(*liste*)

INPUT: Folge von Zahlen, $liste = a_1, ..., a_n$, $a_i \in \mathbb{N}$
if n <=1 **then**
  gebe *liste* zurück
else
  $linkeListe = a_1, ..., a_{\lceil \frac{n}{2} \rceil}$
  $rechteListe = a_{\lceil \frac{n}{2} \rceil + 1}, ..., a_n$
  gebe Sort(SortList(*linkeListe*),SortList(*rechteListe*)) zurück
end if

Algorithm 5 Sort(*rechteListe*, *linkeListe*)

INPUT: zwei Folgen von Zahlen:
$rechteListe = a_1, ..., a_l$, $linkeListe = b_1, ..., b_k$, $a_i, b_i \in \mathbb{N}$
neueListe
while *rechteListe* und *linkeListe* nicht leer **do**
  if erstes Element von *linkeListe* <= erstes Element von *rechteListe* **then**
    füge erstes Element von *linkeListe* hinten in *neueListe* ein und entferne es aus *linkeListe*
  else
    füge erstes Element von *rechteListe* hinten in *neueListe* ein und entferne es aus *rechteListe*
  end if
end while
while *linkeListe* nicht leer **do**
  füge erstes Element von *linkeListe* hinten in *neueListe* ein und entferne es aus *linkeListe*
end while
while *rechteListe* nicht leer **do**
  füge erstes Element von *rechteListe* hinten in *neueListe* ein und entferne es aus *linkeListe*
end while
gib *neueListe* zurück

(a) Sortieren Sie mit *SortList* die Folge 9,10,7,3,1,2,12,9,23

(b) Bestimmen Sie die Laufzeit von *SortList*.

Aufgabe H4 (4 Punkte)

Gegeben sei Algorithmus 6. Was tut der Algorithmus? Bestimmen Sie seine Laufzeit.

Algorithm 6

```
INPUT : n ∈ ℕ
K1 = 2;
K2 = n;
while K2 > K1 do
  K2 = n/K1
  if ⌈K2⌉ == K2 then
    return K1
  else
    K1=K1+1
  end if
end while
return 0
```

Aufgabe H5 (Komplexitätsklassen) (6 Punkte)

Beweisen Sie die folgenden Aussagen für Funktion $f, t : \mathbb{N} \to \mathbb{R}$.

(a) $\mathcal{O}(f) + \mathcal{O}(f) \subseteq \mathcal{O}(f)$,

(b) $\mathcal{O}(f) \cdot \mathcal{O}(t) \subseteq \mathcal{O}(f \cdot t)$,

(c) $\max\{f, t\} \in \Theta(f + t)$ für $f, t \geq 0$.

Die Lösungen zu den Hausübungen geben Sie bitte am *Mittwoch*, den *12.05.2010*, zwischen *7:50 Uhr* und *9:45* in Raum S2|15 211 ab.