

8. Übung zur Vorlesung

COMPUTERORIENTIERTE MATHEMATIK I

WS 2020/2021

[http://numerik.mi.fu-berlin.de/wiki/WS\\_2020/CoMaI.php](http://numerik.mi.fu-berlin.de/wiki/WS_2020/CoMaI.php)

**Abgabe: Do., 28. Januar 2021, 12:15 Uhr**

**1. Aufgabe** (4 Bonus TP)

Sei  $\text{ggT}(a, b)$  der größte gemeinsame Teiler zweier positiver natürlicher Zahlen  $a$  und  $b$ . Zeigen Sie dass

$$\text{ggT}(a, b) = \text{ggT}(b, a \bmod b).$$

**2. Aufgabe** (4 TP)

Im Folgenden betrachten wir die  $\mathcal{O}$ - sowie die  $o$ -Notation stets für  $x \rightarrow \infty$ . Seien  $\alpha, \beta \in \mathbb{R}$  und  $g_1, g_2 : \mathbb{R} \rightarrow \mathbb{R}$  mit  $g_1(x) \neq 0 \neq g_2(x)$  für alle  $x \in \mathbb{R}$ . Beweisen oder widerlegen Sie:

- a)  $x^\alpha \in \mathcal{O}(x^\beta) \Leftrightarrow \alpha \leq \beta$
- b)  $x^\alpha \in o(x^\beta) \Leftrightarrow \alpha < \beta$
- c)  $f_1 \in \mathcal{O}(g_1(x)) \wedge f_2 \in \mathcal{O}(g_2(x)) \Leftrightarrow f_1 + f_2 \in \mathcal{O}(|g_1(x)| + |g_2(x)|)$
- d)  $f_1 \in \mathcal{O}(g_1(x)) \wedge f_2 \in \mathcal{O}(g_2(x)) \Rightarrow f_1 \cdot f_2 \in \mathcal{O}(g_1(x) \cdot g_2(x))$

**3. Aufgabe** (4 TP)

Für  $x \in \mathbb{R}$  und  $n \in \mathbb{N}$  betrachten wir den folgenden Algorithmus, der hier in Form eines Pseudocodes dargestellt ist:

1:  $y = 1$

2: **for**  $k$  **in**  $\text{range}(1, n+1)$ :

3:      $y = y + \frac{x^k}{k!}$

Hierbei werden  $x^k$  und  $k!$  in jedem Schleifendurchlauf berechnet durch

$$x^k = x \cdot x \cdot \dots \cdot x, \quad k! = k \cdot (k-1) \cdot \dots \cdot 2 \cdot 1.$$

- a) Beweisen Sie, dass der Aufwand dieses Algorithmus in  $\mathcal{O}(n^2)$  für  $n \rightarrow \infty$  ist. Mit Aufwand ist hier die Anzahl der nötigen arithmetischen Grundoperationen  $+$ ,  $-$ ,  $\cdot$ ,  $/$  gemeint.

- b) Geben Sie einen alternativen Algorithmus an, der denselben Wert  $y$  berechnet, dessen Aufwand aber in  $\mathcal{O}(n)$  für  $n \rightarrow \infty$  liegt. Beweisen Sie auch diese Aussage.

#### 4. Aufgabe (8 PP + 4 Bonus TP)

Wir wollen die Laufzeiten verschiedener Algorithmen zur Bestimmung von  $\text{ggT}(a, b)$  zweier positiver natürlicher Zahlen  $a, b$  testen. Gehen Sie dafür wie folgt vor:

- a) Implementieren Sie den TumbGGT-Algorithmus aus der Vorlesung als eine Funktion

`ggT_tumb(a,b).`

Hierbei soll ein Tupel  $(c, k)$  zurückgegeben werden, wobei  $c$  der größte gemeinsame Teiler von  $a$  und  $b$  sei und  $k$  die Anzahl der im Algorithmus ausgeführten Divisionen sei. Eine Anwendung von `% (mod)` ist als einfache Division zu werten.

- b) Implementieren Sie analog den Algorithmus TumbGGT++ als Funktion

`ggT_tumbpp(a,b)`

mit Rückgabepaar  $(c, k)$ .

- c) Implementieren Sie analog den euklidischen Algorithmus als Funktion

`ggT_euclid(a,b)`

mit Rückgabepaar  $(c, k)$ .

- d) Schreiben Sie eine Skriptdatei `run8_4.py` mit dem folgenden Ablauf:

- Sei  $n = 1000$ . Es werden zwei Vektoren  $a, b \in \mathbb{R}^n$  mit gleichverteilten Zufallszahlen  $a_i, b_i \in \{100, \dots, 1000\}$  erstellt. Dazu können Sie beispielsweise die Funktion `numpy.random.randint` nutzen.
- Für jeden der obigen drei ggT-Algorithmen wird ein Vektor  $k \in \mathbb{R}^n$  erstellt, so dass  $k_i$  die Anzahl der nötigen Divisionen ist, wenn man den jeweiligen Algorithmus für die Eingabedaten  $a_i$  und  $b_i$  anwendet.
- Im Anschluss an die Berechnung von  $k$  wird ein Histogramm für die Häufigkeit der verschiedenen  $k_i$  geplottet, in dessen Titel der Name des jeweiligen Algorithmus sowie die Werte von `kmin= min1,...,n ki` und `kmax= max1,...,n ki` genannt werden.
- Speichern Sie die Plots jeweils in den Dateien `hist_tumb.png`, `hist_tumbpp.png` und `hist_euclid.png` ab.

- e) Geben Sie für jedes der drei Verfahren theoretische untere und obere Schranken für die Werte von  $\min_i k_i$  und  $\max_i k_i$  an. Beschreiben Sie außerdem Ihre Beobachtungen aus den erzeugten Daten und interpretieren Sie die Histogramme hinsichtlich der theoretischen Schranken. Schreiben Sie Ihre Antwort in die Datei `beobachtungen.txt`.

### 5. Bonusaufgabe (Quiz) (1 Bonus TP/PP)

Formulieren Sie eine Frage zur Vorlesung. Falls Sie die Antwort wissen, geben Sie die richtige Antwort und 3 falsche Antwortmöglichkeiten an.

#### ALLGEMEINE HINWEISE

Die Punkte unterteilen sich in Theoriepunkte (TP) und Programmierpunkte (PP). Bitte beachten Sie die auf der Vorlesungshomepage angegebenen Hinweise zur Bearbeitung und Abgabe der Übungszettel, insbesondere der Programmieraufgaben.

Von: Sina Jrmak  
Jonny Lam  
Thore Brehmer

**Definition:** Landau-Symbol  $O(\cdot)$  für  $f(n), g(n) \rightarrow \infty$  für  $n \rightarrow \infty$

$$\Leftrightarrow \exists c > 0 \exists n_0 > 0 \forall n \geq n_0: g(n) \leq c \cdot f(n)$$

$$f(n) = O(g(n)) \quad \text{für } n \rightarrow \infty \quad \Leftrightarrow \quad \limsup_{n \rightarrow \infty} \frac{f(n)}{g(n)} < \infty$$

# 1. Aufgabe

Brehmer  
Lam  
Itmat

$$\text{z.z.: } \text{ggT}(a, b) = \text{ggT}(b, a \bmod b)$$

$$([a]_b := a \bmod b)$$

Wenn  $(a, b)$  die gleichen gemeinsamen Teiler haben wie  $([a]_b, b)$  stimmt auch der größte gemeinsame Teiler überein.

$$\text{Es genügt z.z.: } g|a \wedge g|b \Leftrightarrow g|b \wedge g|[a]_b \quad \text{für } g \in \mathbb{N}$$

$$\text{"} \Rightarrow \text{"} \cdot g|b - \text{trivial}$$

$$\cdot [a]_b \stackrel{\text{Def. VL}}{=} a - \lfloor a/b \rfloor \cdot b \stackrel{g|a}{=} a' \cdot g - \lfloor a/b \rfloor \cdot b \quad (a' \in \mathbb{N})$$

$$\stackrel{g|b}{=} a' \cdot g - \lfloor a/b \rfloor \cdot b' \cdot g \quad (b' \in \mathbb{N})$$

$$= (a' - \lfloor a/b \rfloor \cdot b') \cdot g$$

$$\underbrace{\begin{matrix} \wedge & \wedge & \wedge \\ \mathbb{N} & \mathbb{N} & \mathbb{N} \end{matrix}}_{\in \mathbb{N}}$$

$$\Rightarrow g|[a]_b$$

$$\text{"} \Leftarrow \text{"} \cdot g|b - \text{trivial}$$

$$\cdot a \stackrel{\text{Def. VL}}{=} [a]_b + \lfloor a/b \rfloor \cdot b \stackrel{g|[a]_b}{=} a' \cdot g + \lfloor a/b \rfloor \cdot b \quad (a' \in \mathbb{N})$$

$$\stackrel{g|b}{=} a' \cdot g + \lfloor a/b \rfloor \cdot b' \cdot g \quad (b' \in \mathbb{N})$$

$$= (a' + \lfloor a/b \rfloor \cdot b') \cdot g$$

$$\underbrace{\begin{matrix} \wedge & \wedge & \wedge \\ \mathbb{N} & \mathbb{N} & \mathbb{N} \end{matrix}}_{\in \mathbb{N}}$$

$$\Rightarrow g|a$$

$$\Rightarrow g|a \wedge g|b \Leftrightarrow g|b \wedge g|[a]_b$$

□

## 2. Aufgabe (4 TP)

Im Folgenden betrachten wir die  $\mathcal{O}$ - sowie die  $o$ -Notation stets für  $x \rightarrow \infty$ . Seien  $\alpha, \beta \in \mathbb{R}$  und  $g_1, g_2: \mathbb{R} \rightarrow \mathbb{R}$  mit  $g_1(x) \neq 0 \neq g_2(x)$  für alle  $x \in \mathbb{R}$ . Beweisen oder widerlegen Sie:

a)  $x^\alpha \in \mathcal{O}(x^\beta) \Leftrightarrow \alpha \leq \beta$

b)  $x^\alpha \in o(x^\beta) \Leftrightarrow \alpha < \beta$

c)  $f_1 \in \mathcal{O}(g_1(x)) \wedge f_2 \in \mathcal{O}(g_2(x)) \Leftrightarrow f_1 + f_2 \in \mathcal{O}(|g_1(x)| + |g_2(x)|)$

d)  $f_1 \in \mathcal{O}(g_1(x)) \wedge f_2 \in \mathcal{O}(g_2(x)) \Rightarrow f_1 \cdot f_2 \in \mathcal{O}(g_1(x) \cdot g_2(x))$

a)  $x^\alpha \in \mathcal{O}(x^\beta) \Leftrightarrow \alpha \leq \beta$

1.  $x^\alpha \in \mathcal{O}(x^\beta) \stackrel{?}{\Leftrightarrow} \alpha \leq \beta$

$$x^\alpha \in \mathcal{O}(x^\beta) \Leftrightarrow \limsup_{x \rightarrow \infty} \frac{x^\alpha}{x^\beta} < \infty$$

$$= \lim_{x \rightarrow \infty} \left( \sup \left\{ \frac{x^\alpha}{x^\beta}, \frac{(x+1)^\alpha}{(x+1)^\beta}, \dots \right\} \right) < \infty$$

$$\stackrel{\text{da } \alpha \leq \beta}{=} \lim_{x \rightarrow \infty} \left( \sup \left\{ \frac{1}{x^{\alpha-\beta}}, \frac{1}{(x+1)^{\alpha-\beta}}, \dots \right\} \right) < \infty$$

$$= \lim_{x \rightarrow \infty} \frac{1}{x^{\alpha-\beta}}$$

für  $\alpha - \beta = 0$ :

$$= 1 < \infty$$

für  $\alpha - \beta > 0$ :

$$= 0 < \infty$$

2.  $x^\alpha \in \mathcal{O}(x^\beta) \stackrel{?}{\Rightarrow} \alpha \leq \beta$

$$x^\alpha \in \mathcal{O}(x^\beta) \Leftrightarrow \limsup_{x \rightarrow \infty} \frac{x^\alpha}{x^\beta} < \infty$$

$$= \lim_{x \rightarrow \infty} \left( \sup \left\{ \frac{x^\alpha}{x^\beta}, \frac{(x+1)^\alpha}{(x+1)^\beta}, \dots \right\} \right) < \infty$$

$$= \lim_{x \rightarrow \infty} \left( \sup \left\{ \frac{1}{x^{\alpha-\beta}}, \frac{1}{(x+1)^{\alpha-\beta}}, \dots \right\} \right) < \infty$$

$$= \lim_{x \rightarrow \infty} \frac{1}{x^{\alpha-\beta}} < \infty$$

für  $\alpha > \beta$

$$\infty < \infty \quad \text{falsch}$$

$$\Rightarrow \alpha \leq \beta$$

für  $\alpha = \beta$ :

$$= 1 < \infty \quad \text{wahr}$$

für  $\alpha < \beta$ :

$$= 0 < \infty \quad \text{wahr}$$

## 2. Aufgabe (4 TP)

Im Folgenden betrachten wir die  $\mathcal{O}$ - sowie die  $o$ -Notation stets für  $x \rightarrow \infty$ . Seien  $\alpha, \beta \in \mathbb{R}$  und  $g_1, g_2: \mathbb{R} \rightarrow \mathbb{R}$  mit  $g_1(x) \neq 0 \neq g_2(x)$  für alle  $x \in \mathbb{R}$ . Beweisen oder widerlegen Sie:

a)  $x^\alpha \in \mathcal{O}(x^\beta) \Leftrightarrow \alpha \leq \beta$

b)  $x^\alpha \in o(x^\beta) \Leftrightarrow \alpha < \beta$

c)  $f_1 \in \mathcal{O}(g_1(x)) \wedge f_2 \in \mathcal{O}(g_2(x)) \Leftrightarrow f_1 + f_2 \in \mathcal{O}(|g_1(x)| + |g_2(x)|)$

d)  $f_1 \in \mathcal{O}(g_1(x)) \wedge f_2 \in \mathcal{O}(g_2(x)) \Rightarrow f_1 \cdot f_2 \in \mathcal{O}(g_1(x) \cdot g_2(x))$

a) z.z.  $x^\alpha \in \mathcal{O}(x^\beta) \Leftrightarrow \alpha \leq \beta$

" $\Rightarrow$ ": Sei  $f(x) = x^\alpha$   
 $g(x) = x^\beta$

Es gilt:  $\limsup_{x \rightarrow \infty} \frac{f(x)}{g(x)} = \limsup_{x \rightarrow \infty} \frac{x^\alpha}{x^\beta} < \infty$

Es muss gelten:  $\alpha \leq \beta$ , sonst:  $\alpha - \beta \geq 1$

$\limsup_{x \rightarrow \infty} \frac{x^\alpha}{x^\beta} = \limsup_{x \rightarrow \infty} x^{\alpha-\beta} \rightarrow \infty$  q.e.d.

" $\Leftarrow$ ":

Angenommen es gilt:  $\alpha > \beta$ , dann folgt:

$\limsup_{x \rightarrow \infty} \frac{x^\alpha}{x^\beta} = \limsup_{x \rightarrow \infty} \frac{1}{x^{\beta-\alpha}} < \infty$  q.e.d.

b)

" $\Rightarrow$ ": Sei  $f(x) = x^\alpha$   
 $g(x) = x^\beta$

Es gilt:  $\lim_{x \rightarrow \infty} \frac{f(x)}{g(x)} = \lim_{x \rightarrow \infty} \frac{x^\alpha}{x^\beta} = 0$

Widerspruch:

Sei  $\alpha = \beta$ , dann gilt:  $\lim_{x \rightarrow \infty} \frac{x^\alpha}{x^\beta} = 1$



c)

Widerspruch:

Sei:  $f_1(x) = x^3$

$f_2(x) = x$

$g_1(x) = x^2$

$g_2(x) = x^3$

Es gilt:

$$f_1 + f_2 \in O(|g_1(x)| + |g_2(x)|) \Leftrightarrow \limsup_{n \rightarrow \infty} \frac{f_1 + f_2}{|g_1 + g_2|}$$

$$\Leftrightarrow \limsup_{n \rightarrow \infty} \frac{x^3 + x}{x^2 + x^3} \Leftrightarrow \limsup_{n \rightarrow \infty} \frac{1}{x} < \infty$$

aber es gilt nicht:

$$f_1 \in O(g_1(x)) \Leftrightarrow \limsup_{x \rightarrow \infty} \frac{f_1}{g_1} \Leftrightarrow \limsup_{x \rightarrow \infty} \frac{x^3}{x^2}$$

$$\Leftrightarrow \limsup_{x \rightarrow \infty} x \neq \infty \quad \downarrow$$

d)

Wir wissen:

$$f_1 \in O(g_1(x)) \Leftrightarrow \limsup_{x \rightarrow \infty} \frac{f_1(x)}{g_1(x)} < \infty$$

$$f_2 \in O(g_2(x)) \Leftrightarrow \limsup_{x \rightarrow \infty} \frac{f_2(x)}{g_2(x)} < \infty$$

also muss  $f_1(x) \leq g_1(x)$  und  $f_2(x) \leq g_2(x)$  für  $x \rightarrow \infty$  gelten, da sonst „ $< \infty$ “ nicht mehr gelten würde, wenn der Zähler größer als der Nenner ist.

$$\text{z.z.: } \limsup_{x \rightarrow \infty} \frac{f_1(x) \cdot f_2(x)}{g_1(x) \cdot g_2(x)} < \infty$$

d.h. wir müssen zeigen:  $f_1(x) \cdot f_2(x) \leq g_1(x) \cdot g_2(x)$ .

Offensichtlich gilt  $f_1(x) \cdot f_2(x) \leq g_1(x) \cdot g_2(x)$ , da wir

$f_1(x) \leq g_1(x)$  und  $f_2(x) \leq g_2(x)$  haben mit  $g_1(x)$  und  $g_2(x) \neq 0$ .

### 3. Aufgabe (4 TP)

Für  $x \in \mathbb{R}$  und  $n \in \mathbb{N}$  betrachten wir den folgenden Algorithmus, der hier in Form eines Pseudocodes dargestellt ist:

1:  $y = 1$

2: for  $k$  in range(1,  $n+1$ ):

3:  $y = y + \frac{x^k}{k!}$

Hierbei werden  $x^k$  und  $k!$  in jedem Schleifendurchlauf berechnet durch

$$x^k = x \cdot x \cdot \dots \cdot x, \quad k! = k \cdot (k-1) \cdot \dots \cdot 2 \cdot 1.$$

- a) Beweisen Sie, dass der Aufwand dieses Algorithmus in  $\mathcal{O}(n^2)$  für  $n \rightarrow \infty$  ist. Mit Aufwand ist hier die Anzahl der nötigen arithmetischen Grundoperationen  $+$ ,  $-$ ,  $\cdot$ ,  $/$  gemeint.

Laufzeit  
(1)

(n)

(1+n+1+n)

$y = 1$

for  $k$  in range (1,  $n+1$ ):

$y = y + \frac{x^k}{k!}$

$+ = 1$

$\leftarrow \text{pow} = n$

$\leftarrow / = 1$

$\leftarrow \text{fak} = n$

}  $n$  mal durch  
for schleife

$$\Rightarrow 1 + n \cdot (2n+2) \stackrel{?}{\in} \mathcal{O}(n^2)$$

$$\lim_{n \rightarrow \infty} \sup \left( \frac{2n^2 + 2n + 1}{n^2} \right) \stackrel{?}{<} \infty$$

$$= \lim_{n \rightarrow \infty} \left( \sup \left\{ 2 + \frac{2}{n} + \frac{1}{n^2}, 2 + \frac{2}{n+1} + \frac{1}{(n+1)^2}, \dots \right\} \right) \stackrel{?}{<} \infty$$

$$= \lim_{n \rightarrow \infty} 2 + \frac{2}{n} + \frac{1}{n^2} \stackrel{?}{<} \infty$$

$$= 2 < \infty \quad \checkmark$$



- b) Geben Sie einen alternativen Algorithmus an, der denselben Wert  $y$  berechnet, dessen Aufwand aber in  $\mathcal{O}(n)$  für  $n \rightarrow \infty$  liegt. Beweisen Sie auch diese Aussage.

Laufzeit

(1)

$$y = 1$$

(1)

$$x\_pow\_k = 1$$

(1)

$$k\_fak = 1$$

(n)

for  $k$  in range  $(1, n+1)$ :

(1)

$$x\_pow\_k *= x\_pow\_k$$

(1)

$$k\_fak *= k$$

(1)

$$y = y + x\_pow\_k / k\_fak$$

$$\Rightarrow (1+1+1+n \cdot (1+1+1)) \stackrel{?}{\in} \mathcal{O}(n)$$

$$\lim_{n \rightarrow \infty} \sup \left( \frac{3n+3}{n} \right) \stackrel{?}{<} \infty$$

$$\lim_{n \rightarrow \infty} \left( \sup \left\{ \frac{3n+3}{n}, \frac{3(n+1)+3}{n+1}, \dots \right\} \right) \stackrel{?}{<} \infty$$

$$\lim_{n \rightarrow \infty} \left( \sup \left\{ 3 + \frac{3}{n}, 3 + \frac{3}{n+1}, \dots \right\} \right) \stackrel{?}{<} \infty$$

$$\lim_{n \rightarrow \infty} 3 + \frac{3}{n} \stackrel{?}{<} \infty$$

$$3 < \infty \quad \checkmark$$