

2.2 (1P)

$$a) \lim_{n \rightarrow \infty} \frac{(n^2 - n + 1)^2}{n^4} = \lim_{n \rightarrow \infty} \frac{n^4 - 2n^3 + 3n^2 - 2n + 1}{n^4} = \lim_{n \rightarrow \infty} 1 - \frac{2}{n} + \frac{3}{n^2} - \frac{2}{n^3} + \frac{1}{n^4} = 1 \in \mathbb{R}^+$$

$\Rightarrow$ Nach Theorem 1 gilt: $(n^2 - n + 1)^2 \leq O(n^4)$ und $n^4 \leq O((n^2 - n + 1)^2)$

$$b) \lim_{n \rightarrow \infty} \frac{\sqrt{n}}{\sqrt[3]{n \cdot \log(n)}} = \lim_{n \rightarrow \infty} \frac{\sqrt{n}}{\sqrt[3]{n} \cdot \sqrt[3]{\log(n)}} = \lim_{n \rightarrow \infty} \frac{n^{\frac{1}{2}}}{n^{\frac{1}{3}} \cdot \log^{\frac{1}{3}} n} = \lim_{n \rightarrow \infty} \frac{n^{\frac{1}{6}}}{\log^{\frac{1}{3}} n} = \infty$$

"Sandwich Theorem"

$$\frac{n^{\frac{1}{6}}}{\log^{\frac{1}{3}} n} \geq \frac{n^{\frac{1}{6}}}{\log n} \stackrel{L'H}{=} \frac{\frac{1}{6} \cdot n^{-\frac{5}{6}}}{\frac{1}{n}} = \frac{\frac{1}{6} n^{\frac{1}{6}}}{\frac{1}{n}} = \frac{1}{6} n^{\frac{7}{6}} \rightarrow \infty$$

$\Rightarrow$ Also: $\sqrt{n} \not\leq O(\sqrt[3]{n \log n})$ (Theorem 1)

$$c) \text{ Wir sehen: } \log_{100}(n) = \frac{\log_e(n)}{\log_e(100)} = \frac{\ln n}{\ln 100} \text{ und } \log_2(n^{100}) = 100 \cdot \log_2(n) = \frac{100 \cdot \ln(n)}{\ln(2)}$$

$$\Rightarrow \lim_{n \rightarrow \infty} \frac{\log_{100}^2(n)}{\log_2(n^{100})} = \lim_{n \rightarrow \infty} \frac{\left(\frac{\ln(n)}{\ln(100)}\right)^2}{100 \cdot \left(\frac{\ln(n)}{\ln(2)}\right)} = \lim_{n \rightarrow \infty} \frac{\left(\frac{\ln^2 n}{\ln^2 100}\right)}{100 \cdot \left(\frac{\ln(n)}{\ln(2)}\right)} = \lim_{n \rightarrow \infty} \frac{\ln^2 n \cdot \ln(2)}{100 \cdot \ln(n) \cdot \ln^2 100}$$

$$= \lim_{n \rightarrow \infty} \frac{\ln(2) \cdot \ln(n)}{100 \cdot \ln^2 100} = \frac{\ln(2)}{100 \cdot \ln^2 100} \cdot \left(\lim_{n \rightarrow \infty} \ln(n)\right) = \infty$$

$\Rightarrow \log_{100}^2(n) \not\leq O(\log_2(n^{100}))$ (Theorem 1)

$$d) \sum_{k=1}^n (k^2 e^k + \ln^3 k) \leq O(3^n) ?$$

Nach oben abschätzen: $\sum_{k=1}^n (k^2 e^k + \ln^3 k) \leq \sum_{k=1}^n (n^2 e^n + \ln^3 n) = n \cdot (n^2 e^n + \ln^3 n)$

Nun wenden wir Theorem 1 an:

$$\lim_{n \rightarrow \infty} \frac{n^3 e^n + n \ln^3 n}{3^n} = \lim_{n \rightarrow \infty} \left(n^3 \cdot \left(\frac{e}{3}\right)^n + \frac{n \cdot \ln^3(n)}{3^n} \right)$$

Offensichtlich gilt: $\lim_{n \rightarrow \infty} \left(\frac{n \cdot \ln^3(n)}{3^n} \right) = 0$ $\left(\frac{n \cdot \ln^3(n)}{3^n} \leq \frac{n \cdot n}{3^n} \leq \frac{2^n}{3^n} = \left(\frac{2}{3}\right)^n \xrightarrow{n \rightarrow \infty} 0 \right)$

Und: $\lim_{n \rightarrow \infty} \left(n^3 \cdot \left(\frac{e}{3}\right)^n \right) = \lim_{n \rightarrow \infty} \left(\frac{n^3}{\left(\frac{3}{e}\right)^n} \right) \stackrel{L'H}{=} \dots = \lim_{n \rightarrow \infty} \frac{6}{(\text{irgendwas exponentiell})} = 0$

Wiel: $\frac{e^n}{3^n} = \frac{1}{\left(\frac{3}{e}\right)^n}$

Hier sagt Musterlösung schon $= 0$

$$\Rightarrow \sum_{k=1}^n (k^2 e^k + \ln^3 k) \leq O(3^n)$$

$$c) \sum_{k=0}^n 2^k \leq O(2^n)$$

Cooler Beweis:

$$\text{Wir sehen: } \sum_{k=0}^n 2^k = 2^0 + 2^1 + \dots + 2^n \quad \text{Also: } 2 \cdot \sum_{k=0}^n 2^k = 2 + 2^2 + \dots + 2^{n+1}$$

$$\Rightarrow \sum_{k=0}^n 2^k - 2 \cdot \sum_{k=0}^n 2^k = (2^0 + 2^1 + 2^2 + \dots + 2^n) - (2^1 + 2^2 + \dots + 2^n + 2^{n+1})$$

$$= 2^0 - 2^{n+1}$$

$$\Leftrightarrow (1-2) \cdot \sum_{k=0}^n 2^k = 2^0 - 2^{n+1}$$

$$\Leftrightarrow \sum_{k=0}^n 2^k = \frac{1-2^{n+1}}{(1-2)} = 2^{n+1} - 1$$

$$\text{Also: } \lim_{n \rightarrow \infty} \frac{\sum_{k=0}^n 2^k}{2^n} = \lim_{n \rightarrow \infty} \frac{2^{n+1} - 1}{2^n} = \lim_{n \rightarrow \infty} \left(2 - \frac{1}{2^n} \right) = 2 \in \mathbb{R}^+$$

$$\Rightarrow \text{Also: } \sum_{k=0}^n 2^k \leq O(2^n)$$

(2.3) Alle Beweisschritte Frage:

$$\text{Show: } n \cdot \ln(n) \leq O(\ln(n!))$$

Achtung, das fragt nicht, ob:
 $n \cdot \ln(n) \leq \ln(n!)$

↑ Das wäre falsch

$$\text{Aus 1.3.c.3 wissen wir: } n! \geq \left(\frac{n}{2}\right)^{\frac{n}{2}} \quad \text{Da Logarithmus monoton wachsend:}$$

$$\ln(n!) \geq \ln\left(\left(\frac{n}{2}\right)^{\frac{n}{2}}\right) = \frac{n}{2} \cdot \ln\left(\frac{n}{2}\right) = \frac{n}{2} \cdot (\ln(n) - \ln(2))$$

$$\Rightarrow \lim_{n \rightarrow \infty} \frac{n \cdot \ln(n)}{\frac{1}{2} n \cdot (\ln(n) - \ln(2))} = \lim_{n \rightarrow \infty} \frac{2 \ln(n)}{\ln(n) - \ln(2)} \stackrel{\text{L'H}}{=} \lim_{x \rightarrow \infty} 2 \cdot \frac{\frac{1}{x}}{\frac{1}{x}} = 2$$

$$\Rightarrow n \cdot \ln(n) \leq O(\ln(n!))$$

2.4 (1P)

- Algorithmen "iterativ" $\Rightarrow$ Machen etwas n mal ↑ normalerweise abhängig von Input-Size
(größer Input $\Rightarrow$ mehr Iterationen)

- Annahme: "later iterations take more time": If $t(i)$ is time taken by i -th iteration then: $t(k) \leq t(m) \quad \forall k < m$

$\hookrightarrow$ Total Runtime: $T(n) = \sum_{k=1}^n t(k)$

- Gucken uns verschiedene $t: \mathbb{N} \rightarrow \mathbb{N}$ an

a) t non-decreasing: _____ oder _____

Show $T(n) \leq O(n \cdot t(n))$

We have: $T(n) = \sum_{k=1}^n t(k) \leq \sum_{k=1}^n t(n) = n \cdot t(n)$ (da t non-decreasing)

$\Rightarrow T(n) \leq O(n \cdot t(n))$

b) t grows polynomially: $\exists \beta > 0, C \geq 1, \beta, C \in \mathbb{R}^+ \cdot \forall n \in \mathbb{N} : \frac{1}{C} \cdot n^\beta \leq t(n) \leq C \cdot n^\beta$

Show: $n \cdot t(n) \leq O(T(n))$ (Hint: For $n \geq 2$: $\sum_{k=\lceil \frac{n}{2} \rceil}^n k^\beta \geq \frac{n}{2} \cdot \left(\frac{n}{2}\right)^\beta$)

Solution: $\forall n \geq 2$:

$$n \cdot t(n) \leq n \cdot (C \cdot n^\beta) = \left(\frac{2}{2}n\right) \cdot C \cdot \left(\frac{2}{2}n\right)^\beta = \left(\frac{n}{2}\right) \cdot C \cdot \left(\frac{n}{2}\right)^\beta \cdot 2^{\beta+1}$$

$$= 2^{\beta+1} \cdot C \cdot \left(\left(\frac{n}{2}\right) \cdot \left(\frac{n}{2}\right)^\beta\right) \stackrel{\text{(Hint)}}{\leq} 2^{\beta+1} \cdot C \cdot \left(\sum_{k=\lceil \frac{n}{2} \rceil}^n k^\beta\right)$$

$$\leq 2^{\beta+1} \cdot C \cdot \left(\sum_{k=\lceil \frac{n}{2} \rceil}^n C \cdot t(k)\right) \leq 2^{\beta+1} \cdot C^2 \cdot \left(\sum_{k=1}^n t(k)\right) = \underbrace{2^{\beta+1} \cdot C^2}_{\text{Konstant}} \cdot T(n)$$

Warum?
Wollen hint
benutzen können

mehr Summanden

Da $\forall n \in \mathbb{N} : \frac{1}{C} n^\beta \leq t(n)$

$\Leftrightarrow n^\beta \leq C \cdot t(n)$

$\Rightarrow$ Also: $n \cdot t(n) \leq O(T(n))$

c) Show for non-decreasing $t: \mathbb{N} \rightarrow \mathbb{N}$ gilt immer: $t(n) \leq O(T(n))$

Solution: $t(n) \leq \sum_{k=1}^n t(k) = T(n) \Rightarrow t(n) \leq O(T(n))$

d auf nächster Seite

d) Annahme: $t(n)$ wächst exponentiell, also:

$$\exists \alpha > 1, C \geq 1 \in \mathbb{R}: \forall n \in \mathbb{N}: \frac{1}{C} \cdot \alpha^n \leq t(n) \leq C \cdot \alpha^n$$

Hint: Use Ex. 1.2

Show: $T(n) \leq O(t(n))$

$$\begin{aligned} T(n) &= \sum_{k=1}^n t(k) \leq \sum_{k=1}^n C \cdot \alpha^k = C \cdot \sum_{k=1}^n \alpha^k \leq C \cdot \sum_{k=0}^n \alpha^k \stackrel{1.2}{=} C \cdot \left(\frac{\alpha^{n+1} - 1}{\alpha - 1} \right) \\ &\leq \frac{C}{\alpha - 1} \cdot \alpha^{n+1} = \frac{C\alpha}{\alpha - 1} \cdot \alpha^n \leq \underbrace{\frac{C^2 \alpha}{\alpha - 1}}_{\text{Konstant}} \cdot t(n) \end{aligned}$$

$$\Rightarrow T(n) \leq O(t(n))$$

2.5 GCD berechnen. Da $\text{gcd}(m, n) = \text{gcd}(n, m)$ nehmen wir w.l.o.g. (without loss of generality) an, dass $n \geq m$

Simplexer Algo: For $d = m \dots 1$ do
if $(n \% d == 0)$
return d ;
break;
return 1;

a) $T_{\text{simple}}(n, m)$. Define for fixed $n \geq 1$: $\min_{1 \leq m \leq n} T_{\text{simple}}(n, m)$
und $\max_{1 \leq m \leq n} T_{\text{simple}}(n, m)$

Annahme: $0 < m \leq n$

$$\text{Wir sehen: } T_{\text{simple}}(n, m) = m - \text{gcd}(n, m) + 1$$

↳ Warum: Weil wir erst für m testen, dann für $m-1$, dann für $m-2$, $m-3$, etc bis gcd .
(und das sind insgesamt $m - \text{gcd}(n, m) + 1$ Tests)

Case Distinction:

$$\bullet m = n \Rightarrow \text{gcd}(n, m) = m \Rightarrow T_{\text{simple}} = m - m + 1 = \underline{\underline{1}}$$

$$\bullet m = n-1 \Rightarrow \text{gcd}(n, m) = \text{gcd}(n, n-1) \Rightarrow T_{\text{simple}} = m - 1 + 1 = m = \underline{\underline{n-1}}$$

$$\bullet 1 \leq m < n-1 \Rightarrow 1 \leq \text{gcd}(n, m) \leq m \Rightarrow 1 \leq T_{\text{simple}}(n, m) \leq m < \underline{\underline{n-1}}$$

$$\Rightarrow \min_{1 \leq m \leq n} T_{\text{simple}}(n, m) = 1 \quad \text{and} \quad \max_{1 \leq m \leq n} T_{\text{simple}}(n, m) = n-1$$

Newer Algo:

Seien q, r s.d.: $n = q \cdot m + r$

$$\rightarrow r = 0 \Rightarrow \gcd(n, m) = m$$

$$\rightarrow r > 0 \Rightarrow \gcd(n, m) = \gcd(m, r)$$

Daraus bauen wir jetzt Algo:
Wir teilen immer n durch m und
betrachten remainder:

$$\begin{aligned} \gcd(n, m) & \left\{ \begin{array}{l} n = q \cdot m + \underbrace{r}_{>0} \\ = \gcd(m, r) \left\{ \begin{array}{l} m = q' \cdot r + \underbrace{r'}_{>0} \\ = \gcd(r, r') \left\{ \begin{array}{l} r = q'' \cdot r' + \underbrace{r''}_{=0} \\ = \underline{\underline{r'}} \end{array} \right. \end{array} \right. \end{array} \right. \end{aligned}$$

$$\text{z.B.: } \left\{ \begin{array}{l} \gcd(18, 12) \\ = \gcd(12, 6) \\ = \underline{\underline{6}} \end{array} \right\} \left\{ \begin{array}{l} 18 = 1 \cdot 12 + 6 \\ 12 = 2 \cdot 6 + 0 \end{array} \right.$$

Als Algo:

for $i = 1 \dots$ do:

if $n_i = 0$ do:

return n_{i-1}

$$r_i = n_{i-1} \% n_i$$

$$n_{i+1} = r_i$$

remainder operator, e.g. $5 \% 3 = 2$
da $5 = 1 \cdot 3 + 2$

e) Show: $\forall i \geq 2: n_i < \frac{n_{i-2}}{2}$

$\Rightarrow n_i$ is remainder of Euclidean division n_{i-2} by n_{i-1}

$\hookrightarrow$ Also: $n_{i-2} = q_{i-1} \cdot n_{i-1} + \underbrace{n_i}_{(= "r_{i-1}")}$

Case Distinction:

• Case 1: $n_{i-1} \leq \frac{n_{i-2}}{2}$

Then we use the fact that

remainder of Euclidean Division is always smaller than divisor

Hence: $n_i < n_{i-1} \leq \frac{n_{i-2}}{2}$

• Case 2: $n_{i-1} > \frac{n_{i-2}}{2}$, then:

$$n_{i-2} - n_{i-1} < n_{i-2} - \frac{n_{i-2}}{2} = \frac{n_{i-2}}{2} < n_{i-1}$$

Und wir schon: $n_{i-2} = n_{i-1} + (n_{i-2} - n_{i-1})$ mit $(n_{i-2} - n_{i-1}) < n_{i-1}$

$\Rightarrow$ Also muss $q_{i-1} = 1$ und $r_{i-1} = n_{i-2} - n_{i-1}$

Denn:

$$n_{i-2} = q_{i-1} \cdot n_{i-1} + \underbrace{n_i}_{(= "r_{i-1}")}$$

Erklärung:

Wir teilen n_{i-2} euklidisch durch n_{i-1} und wissen: $n_{i-2} = q_{i-1} \cdot n_{i-1} + r_{i-1}$

ABER wir wissen jetzt auch: $n_{i-2} = n_{i-1} + (n_{i-2} - n_{i-1})$

$\Rightarrow$ Koeffizientenvergleich: $q_{i-1} = 1$ und $r_{i-1} = (n_{i-2} - n_{i-1})$

Und da $n_i := r_{i-1}$, gilt:

$$n_i = r_{i-1} = n_{i-2} - n_{i-1} < \frac{n_{i-2}}{2}$$

f) Wir def.: $T_E(n, m) := \# \text{iterations to find } \gcd(n, m)$

Show: $T_E(n, m) \leq O(\log(n))$

Intuition: $n_i < \frac{n_{i-2}}{2}$ (immer!) $\rightarrow$ Weg zu $n_i = 0$ also schnell

Wir können leicht per Induktion zeigen: $n_{2k} < \frac{n_0}{2^k} = \frac{n}{2^k}$

Also: Wenn $k = \lceil \log_2(n) \rceil \geq 1$, dann $n_{2k} < \frac{n}{n} = 1 \Rightarrow n_{2k} = 0$

Also: $T_E(n, m) \leq 2 \cdot \lceil \log_2(n) \rceil \leq 2 \cdot (\log(n) + 1) \leq 4 \cdot \log_2(n) \leq \underline{O(\log(n))}$

Deshalb "2.": Alle 2 Schritte halbieren wir n