Einführung in Algorithmen

Übungen

Aufgabe 1

Warum ist ein Kochrezept strenggenommen kein Algorithmus.

Aufgabe 1

Die Beschreibung ist nicht präzise genug, damit sie von einer Maschine ausgeführt werden kann. Darüber hinaus kann mit einem Kochrezept genau ein Gericht gekocht werden. Ein Algorithmus dient jedoch der Lösung einer ganzen Klasse von Aufgaben.

Aufgabe 2

Zeige schrittweise, wie der klassische Algorithmus von Euklid den $\mathrm{ggT}(17, 5)$ berechnet, indem du die Kette *aller* Zwischenresultate notierst.

Aufgabe 2

$$\begin{aligned}
\mathrm{ggT}(17,5) &= \mathrm{ggT}(5,12) = \mathrm{ggT}(12,5) = \mathrm{ggT}(5,7) = \mathrm{ggT}(7,5) \\
&= \mathrm{ggT}(5,2) = \mathrm{ggT}(2,3) = \mathrm{ggT}(3,2) = \mathrm{ggT}(2,1) \\
&= \mathrm{ggT}(1,1) = \mathrm{ggT}(1,0) = 1
\end{aligned}$$

Aufgabe 3

Zeige, warum die klassische Version des euklidischen Algorithmus nicht funktioniert, wenn eine der beiden Zahlen negativ ist, indem du $\mathrm{ggT}(8, -6)$ berechnest.

Aufgabe 3

$$
\begin{aligned}
\operatorname{ggT}(8, -6) &= \operatorname{ggT}(-6, 14) = \operatorname{ggT}(14, -6) = \operatorname{ggT}(-6, 20) \\
&= \operatorname{ggT}(20, -6) = \operatorname{ggT}(-6, 26) = \operatorname{ggT}(26, -6) \\
&= \dots
\end{aligned}
$$

Der Algorithmus erzeugt immer grössere Werte für a und terminiert daher nicht mehr. Dasselbe geschieht, auch, wenn nach der Differenzenbildung die grössere Zahl nicht an die erste Stelle „getauscht" wird.

Aufgabe 4

Zeige schrittweise, wie der moderne Algorithmus von Euklid den $\mathrm{ggT}(17, 5)$ berechnet, indem du die Kette *aller* Zwischenresultate notierst.

Aufgabe 4

$$\mathrm{ggT}(17,5) = \mathrm{ggT}(5,2) = \mathrm{ggT}(2,1) = \mathrm{ggT}(1,0) = 1$$

Aufgabe 5

Zeige schrittweise, wie der moderne Algorithmus von Euklid den $\text{ggT}(72, 116)$ berechnet, indem du die Kette *aller* Zwischenresultate notierst.

Aufgabe 5

$$
\begin{aligned}
\mathrm{ggT}(72, 116) &= \mathrm{ggT}(116, 72) = \mathrm{ggT}(72, 44) = \mathrm{ggT}(44, 28) \\
&= \mathrm{ggT}(28, 16) = \mathrm{ggT}(16, 12) = \mathrm{ggT}(12, 4) \\
&= \mathrm{ggT}(4, 0) = 4
\end{aligned}
$$

Aufgabe 6

Implementiere den klassischen Algorithmus von Euklid als Python-Funktion mit dem Funktionskopf `euklid_classic(a, b)`. Sorge vor dem Ausführen des Verfahrens dafür, dass beide Argumente `a` und `b` durch Bildung des Absolutbetrags nicht negativ sind.

Aufgabe 6

```
def euklid_classic(a, b):
    a, b = abs(a), abs(b)
    while b != 0:
        if a < b:
            a, b = b, a
        a, b = b, a-b
    return a
```

Aufgabe 7

Implementiere den modernen Algorithmus von Euklid als Python-Funktion mit dem Funktionskopf `euklid_classic(a, b)`. Sorge vor dem Ausführen des Verfahrens dafür, dass beide Argumente `a` und `b` durch Bildung des Absolutbetrags nicht negativ sind.

Aufgabe 7

```
def euklid_modern(a, b):
    a, b = abs(a), abs(b)
    while b != 0:
        a, b = b, a % b
    return a
```

Aufgabe 8

Bestimme für jede Funktion $f(n)$ und jede Problemgrösse n die Dauer t, wenn der Algorithmus $f(n)$ Sekunden zur Lösung des Problems benötigt.

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1				
$\log_2 n$				
$\sqrt{n}$				
n				
$n \log_2 n$				
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1				
$\log_2 n$				
$\sqrt{n}$				
n				
$n \log_2 n$				
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1			
$\log_2 n$				
$\sqrt{n}$				
n				
$n \log_2 n$				
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1		
$\log_2 n$				
$\sqrt{n}$				
n				
$n \log_2 n$				
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	
$\log_2 n$				
$\sqrt{n}$				
n				
$n \log_2 n$				
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$				
$\sqrt{n}$				
n				
$n \log_2 n$				
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1			
$\sqrt{n}$				
n				
$n \log_2 n$				
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2		
$\sqrt{n}$				
n				
$n \log_2 n$				
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	
$\sqrt{n}$				
n				
$n \log_2 n$				
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$				
n				
$n \log_2 n$				
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4			
n				
$n \log_2 n$				
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2		
n				
$n \log_2 n$				
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	
n				
$n \log_2 n$				
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n				
$n \log_2 n$				
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2			
$n \log_2 n$				
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4		
$n \log_2 n$				
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4	8	
$n \log_2 n$				
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4	8	16
$n \log_2 n$				
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4	8	16
$n \log_2 n$	2			
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4	8	16
$n \log_2 n$	2	8		
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4	8	16
$n \log_2 n$	2	8	24	
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4	8	16
$n \log_2 n$	2	8	24	64
n^2				
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4	8	16
$n \log_2 n$	2	8	24	64
n^2	4			
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4	8	16
$n \log_2 n$	2	8	24	64
n^2	4	16		
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n=2$	$n=4$	$n=8$	$n=16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4	8	16
$n \log_2 n$	2	8	24	64
n^2	4	16	64	
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4	8	16
$n \log_2 n$	2	8	24	64
n^2	4	16	64	256
n^3				
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4	8	16
$n \log_2 n$	2	8	24	64
n^2	4	16	64	256
n^3	8			
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4	8	16
$n \log_2 n$	2	8	24	64
n^2	4	16	64	256
n^3	8	64		
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n=2$	$n=4$	$n=8$	$n=16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4	8	16
$n \log_2 n$	2	8	24	64
n^2	4	16	64	256
n^3	8	64	512	
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4	8	16
$n \log_2 n$	2	8	24	64
n^2	4	16	64	256
n^3	8	64	512	4096
2^n				65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4	8	16
$n \log_2 n$	2	8	24	64
n^2	4	16	64	256
n^3	8	64	512	4096
2^n	4			65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4	8	16
$n \log_2 n$	2	8	24	64
n^2	4	16	64	256
n^3	8	64	512	4096
2^n	4	16		65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4	8	16
$n \log_2 n$	2	8	24	64
n^2	4	16	64	256
n^3	8	64	512	4096
2^n	4	16	256	65536
$n!$			40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4	8	16
$n \log_2 n$	2	8	24	64
n^2	4	16	64	256
n^3	8	64	512	4096
2^n	4	16	256	65536
$n!$	2		40320	$2.1 \cdot 10^{13}$

Aufgabe 8

$f(n)$	$n = 2$	$n = 4$	$n = 8$	$n = 16$
1	1	1	1	1
$\log_2 n$	1	2	3	4
$\sqrt{n}$	1.4	2	2.8	4
n	2	4	8	16
$n \log_2 n$	2	8	24	64
n^2	4	16	64	256
n^3	8	64	512	4096
2^n	4	16	256	65536
$n!$	2	24	40320	$2.1 \cdot 10^{13}$

Aufgabe 9

Zu welcher Komplexitätsklasse gehören die Algorithmen mit der folgenden Laufzeitfunktion $T(n)$ [in ms].

(a) $T(n) = 4n + 5n^2 - 2$

(b) $T(n) = 2^{n+3}$

(c) $T(n) = 4$

(d) $T(n) = \sqrt{7.6n}$

(e) $T(n) = \log_2(234n)$

(f) $T(n) = (4n + 3)(5n - 4)(7n - 6)$

Aufgabe 9

(a) $T(n) = 4n + 5n^2 - 2$

Aufgabe 9

(a) $T(n) = 4n + 5n^2 - 2 \in \mathcal{O}(n^2)$

Aufgabe 9

(a) $T(n) = 4n + 5n^2 - 2 \in \mathcal{O}(n^2)$

(b) $T(n) = 2^{n+3} = 2^n \cdot 2^3$

Aufgabe 9

(a) $T(n) = 4n + 5n^2 - 2 \in \mathcal{O}(n^2)$

(b) $T(n) = 2^{n+3} = 2^n \cdot 2^3 \in \mathcal{O}(2^n)$

Aufgabe 9

(a) $T(n) = 4n + 5n^2 - 2 \in \mathcal{O}(n^2)$

(b) $T(n) = 2^{n+3} = 2^n \cdot 2^3 \in \mathcal{O}(2^n)$

(c) $T(n) = 4$

Aufgabe 9

(a) $T(n) = 4n + 5n^2 - 2 \in \mathcal{O}(n^2)$

(b) $T(n) = 2^{n+3} = 2^n \cdot 2^3 \in \mathcal{O}(2^n)$

(c) $T(n) = 4 \in \mathcal{O}(1)$

Aufgabe 9

(a) $T(n) = 4n + 5n^2 - 2 \in \mathcal{O}(n^2)$

(b) $T(n) = 2^{n+3} = 2^n \cdot 2^3 \in \mathcal{O}(2^n)$

(c) $T(n) = 4 \in \mathcal{O}(1)$

(d) $T(n) = \sqrt{7.6n} = \sqrt{7.6} \cdot \sqrt{n}$

Aufgabe 9

(a) $T(n) = 4n + 5n^2 - 2 \in \mathcal{O}(n^2)$

(b) $T(n) = 2^{n+3} = 2^n \cdot 2^3 \in \mathcal{O}(2^n)$

(c) $T(n) = 4 \in \mathcal{O}(1)$

(d) $T(n) = \sqrt{7.6n} = \sqrt{7.6} \cdot \sqrt{n} \in \mathcal{O}(\sqrt{n})$

Aufgabe 9

(a) $T(n) = 4n + 5n^2 - 2 \in \mathcal{O}(n^2)$

(b) $T(n) = 2^{n+3} = 2^n \cdot 2^3 \in \mathcal{O}(2^n)$

(c) $T(n) = 4 \in \mathcal{O}(1)$

(d) $T(n) = \sqrt{7.6n} = \sqrt{7.6} \cdot \sqrt{n} \in \mathcal{O}(\sqrt{n})$

(e) $T(n) = \log_2(234n)$

Aufgabe 9

(a) $T(n) = 4n + 5n^2 - 2 \in \mathcal{O}(n^2)$

(b) $T(n) = 2^{n+3} = 2^n \cdot 2^3 \in \mathcal{O}(2^n)$

(c) $T(n) = 4 \in \mathcal{O}(1)$

(d) $T(n) = \sqrt{7.6n} = \sqrt{7.6} \cdot \sqrt{n} \in \mathcal{O}(\sqrt{n})$

(e) $T(n) = \log_2(234n) = \log_2 234 + \log_2 n \in \mathcal{O}(\log_2 n)$

Aufgabe 9

(a) $T(n) = 4n + 5n^2 - 2 \in \mathcal{O}(n^2)$

(b) $T(n) = 2^{n+3} = 2^n \cdot 2^3 \in \mathcal{O}(2^n)$

(c) $T(n) = 4 \in \mathcal{O}(1)$

(d) $T(n) = \sqrt{7.6n} = \sqrt{7.6} \cdot \sqrt{n} \in \mathcal{O}(\sqrt{n})$

(e) $T(n) = \log_2(234n) = \log_2 234 + \log_2 n \in \mathcal{O}(\log_2 n)$

(f) $T(n) = (4n + 3)(5n - 4)(7n - 6)$

Aufgabe 9

(a) $T(n) = 4n + 5n^2 - 2 \in \mathcal{O}(n^2)$

(b) $T(n) = 2^{n+3} = 2^n \cdot 2^3 \in \mathcal{O}(2^n)$

(c) $T(n) = 4 \in \mathcal{O}(1)$

(d) $T(n) = \sqrt{7.6n} = \sqrt{7.6} \cdot \sqrt{n} \in \mathcal{O}(\sqrt{n})$

(e) $T(n) = \log_2(234n) = \log_2 234 + \log_2 n \in \mathcal{O}(\log_2 n)$

(f) $T(n) = (4n + 3)(5n - 4)(7n - 6) \in \mathcal{O}(n^3)$

Aufgabe 10

In welcher Komplexitätsklasse befindet sich $T_1(n) + T_2(n)$, wenn $T_1(n) \in \mathcal{O}(n^2)$ und $T_2(n) \in \mathcal{O}(n^3)$ gilt?

Aufgabe 10

$$\big(T_1(n) + T_2(n)\big) \in \mathcal{O}\big(\max(n^2, n^3)\big) = \mathcal{O}(n^3)$$

Aufgabe 11

In welcher Komplexitätsklasse befindet sich $T_1(n) \cdot T_2(n)$, wenn $T_1(n) \in \mathcal{O}(n^2)$ und $T_2(n) \in \mathcal{O}(n^3)$ gilt?

Aufgabe 11

$$T_1(n) \cdot T_2(n) \in \mathcal{O}\left(n^2 \cdot n^3\right) = \mathcal{O}(n^5)$$

Aufgabe 12

Eine Implementation eines Algorithmus' hat eine Laufzeitkomplexität von $\mathcal{O}(n^2)$ und benötigt etwa $20\,\mu$s für das Lösen eines Problems der Grösse $n = 100$. Bestimme die ungefähre Laufzeit für ein Problem der Grösse $n = 200$.

Aufgabe 12

$T(n) = C \cdot 100^2 = 20\,\mu\text{s}$ $(*)$ [C ist ein Proportionalitätsfaktor]

Aufgabe 12

$T(n) = C \cdot 100^2 = 20\,\mu s$ $(*)$ [C ist ein Proportionalitätsfaktor]

$$T(2n) = C \cdot 200^2 = C \cdot (2 \cdot 100)^2 = 2^2 \cdot C \cdot 100^2 \overset{(*)}{=} 4 \cdot 20\,\mu s = 80\,\mu s$$

Aufgabe 12

$T(n) = C \cdot 100^2 = 20\,\mu\text{s}$ $(*)$ [C ist ein Proportionalitätsfaktor]

$$T(2n) = C \cdot 200^2 = C \cdot (2 \cdot 100)^2 = 2^2 \cdot C \cdot 100^2 \overset{(*)}{=} 4 \cdot 20\,\mu\text{s} = 80\,\mu\text{s}$$

Allgemein: In $O(n^2)$ bewirkt das Verdoppeln der Problemgrösse eine Vervierfachung der Laufzeit.

Aufgabe 12

$T(n) = C \cdot 100^2 = 20\,\mu\text{s}\ (*)$ [C ist ein Proportionalitätsfaktor]

$$T(2n) = C \cdot 200^2 = C \cdot (2 \cdot 100)^2 = 2^2 \cdot C \cdot 100^2 \stackrel{(*)}{=} 4 \cdot 20\,\mu\text{s} = 80\,\mu\text{s}$$

Allgemein: In $O(n^2)$ bewirkt das Verdoppeln der Problemgrösse eine Vervierfachung der Laufzeit.

Man hätte auch die erste Gleichung nach C auflösen und diesen Wert in die zweite Gleichung einsetzen können. Meist lässt sich die Rechung jedoch in der oben beschriebenen Weise „kurzschliessen“.

Aufgabe 13

Eine Implementation eines Algorithmus' hat eine Laufzeitkomplexität von $\mathcal{O}(\sqrt{n})$ und benötigt etwa 10 ms für das Lösen eines Problems der Grösse $n = 200$. Bestimme die ungefähre Laufzeit für ein Problem der Grösse $n = 20000$.

Aufgabe 13

$$T(200) = C \cdot \sqrt{200} = 10\,\text{ms} \; (*)$$

Aufgabe 13

$$T(200) = C \cdot \sqrt{200} = 10\,\mathrm{ms}\ (*)$$

$$T(20\,000) = C \cdot \sqrt{100 \cdot 200}$$

Aufgabe 13

$$T(200) = C \cdot \sqrt{200} = 10\,\text{ms} \ (*)$$

$$\begin{aligned} T(20\,000) &= C \cdot \sqrt{100 \cdot 200} \\ &= C \cdot \sqrt{100} \cdot \sqrt{200} = 10 \cdot C \cdot \sqrt{200} \end{aligned}$$

Aufgabe 13

$$T(200) = C \cdot \sqrt{200} = 10\,\text{ms} \ (*)$$

$$\begin{aligned} T(20\,000) &= C \cdot \sqrt{100 \cdot 200} \\ &= C \cdot \sqrt{100} \cdot \sqrt{200} = 10 \cdot C \cdot \sqrt{200} \\ &\overset{(*)}{=} 10 \cdot 10\,\text{ms} = 100\,\text{ms} \end{aligned}$$

Aufgabe 13

$$T(200) = C \cdot \sqrt{200} = 10\,\text{ms} \ (*)$$

$$\begin{aligned} T(20\,000) &= C \cdot \sqrt{100 \cdot 200} \\ &= C \cdot \sqrt{100} \cdot \sqrt{200} = 10 \cdot C \cdot \sqrt{200} \\ &\overset{(*)}{=} 10 \cdot 10\,\text{ms} = 100\,\text{ms} \end{aligned}$$

Allgemein: In $O(\sqrt{n})$ bewirkt eine Vergrösserung der Problemgrösse mit dem Faktor 100 eine Vergrösserung der Laufzeit mit dem Faktor $\sqrt{100} = 10$.

Aufgabe 14

Eine Implementation eines Algorithmus' hat eine Laufzeitkomplexität von $\mathcal{O}(\log_2 n)$ und benötigt etwa 5 s für das Lösen eines Problems der Grösse $n = 1000$. Bestimme die ungefähre Laufzeit für ein Problem der Grösse $n = 8\,000$.

Aufgabe 14

$$T(1000) = C \cdot \log_2(1000) = 5\,\mathrm{s}\ (*)$$

Aufgabe 14

$$T(1000) = C \cdot \log_2(1000) = 5\,\mathrm{s}\ (*)$$

$$T(8000) = C \cdot \log_2(8 \cdot 1000) = C\big[\log_2(8) + \log_2(1000)\big]$$

Aufgabe 14

$$T(1000) = C \cdot \log_2(1000) = 5\,\mathrm{s}\ (*)$$

$$\begin{aligned} T(8000) &= C \cdot \log_2(8 \cdot 1000) = C\left[\log_2(8) + \log_2(1000)\right] \\ &= C \log_2(8) + C \log_2(1000) \end{aligned}$$

Aufgabe 14

$$T(1000) = C \cdot \log_2(1000) = 5\,\mathrm{s}\ (*)$$

$$\begin{aligned} T(8000) &= C \cdot \log_2(8 \cdot 1000) = C\left[\log_2(8) + \log_2(1000)\right] \\ &= C\log_2(8) + C\log_2(1000) \\ &\overset{(*)}{=} C \cdot 3 + 5\,\mathrm{s} = \dots \end{aligned}$$

Aufgabe 14

$$T(1000) = C \cdot \log_2(1000) = 5\,\mathrm{s}\ (*)$$

$$\begin{aligned} T(8000) &= C \cdot \log_2(8 \cdot 1000) = C\big[\log_2(8) + \log_2(1000)\big] \\ &= C\log_2(8) + C\log_2(1000) \\ &\overset{(*)}{=} C \cdot 3 + 5\,\mathrm{s} = \dots \end{aligned}$$

Hier benötigen wir den konkreten Wert der Konstanten C.

$C \cdot \log_2(1000) = 5\,\mathrm{s}$ verwende $10^3 \approx 2^{10}$

Aufgabe 14

$$T(1000) = C \cdot \log_2(1000) = 5\,\mathrm{s}\ (*)$$

$$\begin{aligned} T(8000) &= C \cdot \log_2(8 \cdot 1000) = C\big[\log_2(8) + \log_2(1000)\big] \\ &= C\log_2(8) + C\log_2(1000) \\ &\overset{(*)}{=} C \cdot 3 + 5\,\mathrm{s} = \dots \end{aligned}$$

Hier benötigen wir den konkreten Wert der Konstanten C.

$C \cdot \log_2(1000) = 5\,\mathrm{s}$ verwende $10^3 \approx 2^{10}$

$C \cdot \log_2(2^{10}) \approx 5\,\mathrm{s}$

Aufgabe 14

$$T(1000) = C \cdot \log_2(1000) = 5\,\mathrm{s}\ (*)$$

$$\begin{aligned} T(8000) &= C \cdot \log_2(8 \cdot 1000) = C\big[\log_2(8) + \log_2(1000)\big] \\ &= C \log_2(8) + C \log_2(1000) \\ &\overset{(*)}{=} C \cdot 3 + 5\,\mathrm{s} = \dots \end{aligned}$$

Hier benötigen wir den konkreten Wert der Konstanten C.

$$\begin{aligned} C \cdot \log_2(1000) &= 5\,\mathrm{s} \quad \text{verwende } 10^3 \approx 2^{10} \\ C \cdot \log_2(2^{10}) &\approx 5\,\mathrm{s} \\ C \cdot 10 &\approx 5\,\mathrm{s} \\ C &\approx 0.5\,\mathrm{s} \end{aligned}$$

Aufgabe 14

$$T(1000) = C \cdot \log_2(1000) = 5\,\mathrm{s}\ (*)$$

$$\begin{aligned} T(8000) &= C \cdot \log_2(8 \cdot 1000) = C\big[\log_2(8) + \log_2(1000)\big] \\ &= C \log_2(8) + C \log_2(1000) \\ &\overset{(*)}{=} C \cdot 3 + 5\,\mathrm{s} = \ldots \end{aligned}$$

Hier benötigen wir den konkreten Wert der Konstanten C.

$$\begin{aligned} C \cdot \log_2(1000) &= 5\,\mathrm{s} \quad \text{verwende } 10^3 \approx 2^{10} \\ C \cdot \log_2(2^{10}) &\approx 5\,\mathrm{s} \\ C \cdot 10 &\approx 5\,\mathrm{s} \\ C &\approx 0.5\,\mathrm{s} \end{aligned}$$

Damit: $\ldots = 0.5\,\mathrm{s} \cdot 3 + 5\,\mathrm{s} = 6.5\,\mathrm{s}$

Allgemein: Multipliziert man die Problemgrösse eines logarithmisch wachsenden Algorithmus mit dem Faktor k, so erhöht sich die Laufzeit um den Summanden $C \log(k)$.

Aufgabe 15

Eine Implementation eines Algorithmus' hat eine Laufzeitkomplexität von $\mathcal{O}(n!)$ und benötigt etwa 50 ms für das Lösen eines Problems der Grösse $n = 19$. Bestimme die ungefähre Laufzeit für ein Problem der Grösse $n = 20$.

Aufgabe 15

$$T(19) = C \cdot 19! = 50\,\text{ms} \quad (*)$$

Aufgabe 15

$$T(19) = C \cdot 19! = 50\,\text{ms}\ (*)$$

$$T(20) = C \cdot 20!$$

Aufgabe 15

$$T(19) = C \cdot 19! = 50\,\mathrm{ms}\ (*)$$

$$T(20) = C \cdot 20! = C \cdot 20 \cdot 19!$$

Aufgabe 15

$$T(19) = C \cdot 19! = 50\,\mathrm{ms} \; (*)$$

$$T(20) = C \cdot 20! = C \cdot 20 \cdot 19! = 20 \cdot C \cdot 19!$$

Aufgabe 15

$$T(19) = C \cdot 19! = 50\,\text{ms} \; (*)$$

$$T(20) = C \cdot 20! = C \cdot 20 \cdot 19! = 20 \cdot C \cdot 19!$$

$$\overset{(*)}{=} 20 \cdot 50\,\text{ms} = 1000 \;\text{ms} = 1\,\text{s}$$

Aufgabe 15

$$T(19) = C \cdot 19! = 50\,\mathrm{ms}\ (*)$$

$$T(20) = C \cdot 20! = C \cdot 20 \cdot 19! = 20 \cdot C \cdot 19!$$

$$\stackrel{(*)}{=} 20 \cdot 50\,\mathrm{ms} = 1000\ \mathrm{ms} = 1\,\mathrm{s}$$

Allgemein: Vergrössert man ein exponentiell wachsendes Problem der Grösse n um eine weitere Eingabe, so erhöht sich die Laufzeit um den Faktor $n+1$.

Aufgabe 16

Bestimme die Komplexitätsklasse des Python Code-Fragments:

```python
s = 0
for i in range(0, len(A)):
    s += A[i]
```

Aufgabe 16

```
s = 0
for i in range(0, len(A)):
    s += A[i]
```

Zeile	Kosten	Anzahl

Aufgabe 16

```
1 s = 0
2 for i in range(0, len(A)):
3     s += A[i]
```

Zeile	Kosten	Anzahl
1	c_1	1

Aufgabe 16

```
s = 0
for i in range(0, len(A)):
    s += A[i]
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	n

Aufgabe 16

```
s = 0
for i in range(0, len(A)):
    s += A[i]
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	n
3	c_3	n

Aufgabe 16

```python
s = 0
for i in range(0, len(A)):
    s += A[i]
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	n
3	c_3	n

$$T(n) = c_1 \cdot 1 + (c_2 + c_3) \cdot n$$

Aufgabe 16

```
1 s = 0
2 for i in range(0, len(A)):
3     s += A[i]
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	n
3	c_3	n

$$T(n) = c_1 \cdot 1 + (c_2 + c_3) \cdot n \in \mathcal{O}(n) \text{ wobei } n = \texttt{len(A)}$$

Aufgabe 17

Bestimme die Komplexitätsklasse des Python Code-Fragments:

```
s = 1
for i in range(1, n):
    for j in range(1, n):
        s = s + i*j
```

Aufgabe 17

```
s = 1
for i in range(1, n):
    for j in range(1, n):
        s = s + i*j
```

Zeile	Kosten	Anzahl

Aufgabe 17

```
s = 1
for i in range(1, n):
    for j in range(1, n):
        s = s + i*j
```

Zeile	Kosten	Anzahl
1	c_1	1

Aufgabe 17

```
s = 1
for i in range(1, n):
    for j in range(1, n):
        s = s + i*j
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	$n-1$

Aufgabe 17

```
s = 1
for i in range(1, n):
    for j in range(1, n):
        s = s + i*j
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	$n-1$
3	c_3	$(n-1)(n-1)$

Aufgabe 17

```python
s = 1
for i in range(1, n):
    for j in range(1, n):
        s = s + i*j
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	$n-1$
3	c_3	$(n-1)(n-1)$
4	c_4	$(n-1)(n-1)$

Aufgabe 17

```
1 s = 1
2 for i in range(1, n):
3     for j in range(1, n):
4         s = s + i*j
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	$n-1$
3	c_3	$(n-1)(n-1)$
4	c_4	$(n-1)(n-1)$

$$T(n) = c_1 \cdot 1 + c_2(n-1) + (c_3 + c_4)(n-1)(n-1)$$

Aufgabe 17

```
1  s = 1
2  for i in range(1, n):
3      for j in range(1, n):
4          s = s + i*j
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	$n-1$
3	c_3	$(n-1)(n-1)$
4	c_4	$(n-1)(n-1)$

$$T(n) = c_1 \cdot 1 + c_2(n-1) + (c_3 + c_4)(n-1)(n-1) \in \mathcal{O}(n^2)$$

Aufgabe 18

Bestimme die Komplexitätsklasse des Python Code-Fragments.

```
a = 4
b = a**2
c = -b
d = (a+b)*c
```

Aufgabe 18

```
1 a = 4
2 b = a**2
3 c = -b
4 d = (a+b)*c
```

Zeile	Kosten	Anzahl

Aufgabe 18

```
a = 4
b = a**2
c = -b
d = (a+b)*c
```

Zeile	Kosten	Anzahl
1	c_1	1

Aufgabe 18

```
1 a = 4
2 b = a**2
3 c = -b
4 d = (a+b)*c
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	1

Aufgabe 18

```
1 a = 4
2 b = a**2
3 c = -b
4 d = (a+b)*c
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	1
3	c_3	1

Aufgabe 18

```
1 a = 4
2 b = a**2
3 c = -b
4 d = (a+b)*c
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	1
3	c_3	1
4	c_4	1

Aufgabe 18

```
1 a = 4
2 b = a**2
3 c = -b
4 d = (a+b)*c
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	1
3	c_3	1
4	c_4	1

$$T(n) = (c_1 + c_2 + c_3 + c_4) \cdot 1$$

Aufgabe 18

```
1 a = 4
2 b = a**2
3 c = -b
4 d = (a+b)*c
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	1
3	c_3	1
4	c_4	1

$$T(n) = (c_1 + c_2 + c_3 + c_4) \cdot 1 \in \mathcal{O}(1)$$

Aufgabe 19

Bestimme die Komplexitätsklasse des folgenden Code-Fragments:

```
i = n
s = 0
while i > 0:
    s = s + 1
    i = i // 2
```

Aufgabe 19

```python
i = n
s = 0
while i > 0:
    s = s + 1
    i = i // 2
```

Zeile	Kosten	Anzahl

Aufgabe 19

```
i = n
s = 0
while i > 0:
    s = s + 1
    i = i // 2
```

Zeile	Kosten	Anzahl
1	c_1	1

Aufgabe 19

```
1 i = n
2 s = 0
3 while i > 0:
4     s = s + 1
5     i = i // 2
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	1

Aufgabe 19

```
1 i = n
2 s = 0
3 while i > 0:
4     s = s + 1
5     i = i // 2
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	1
3	c_3	$\log_2 n$

Aufgabe 19

```
1 i = n
2 s = 0
3 while i > 0:
4     s = s + 1
5     i = i // 2
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	1
3	c_3	$\log_2 n$
4	c_4	$\log_2 n$

Aufgabe 19

```
1 i = n
2 s = 0
3 while i > 0:
4     s = s + 1
5     i = i // 2
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	1
3	c_3	$\log_2 n$
4	c_4	$\log_2 n$
5	c_5	$\log_2 n$

Aufgabe 19

```
1 i = n
2 s = 0
3 while i > 0:
4     s = s + 1
5     i = i // 2
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	1
3	c_3	$\log_2 n$
4	c_4	$\log_2 n$
5	c_5	$\log_2 n$

Aufgabe 19

```
1 i = n
2 s = 0
3 while i > 0:
4     s = s + 1
5     i = i // 2
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	1
3	c_3	$\log_2 n$
4	c_4	$\log_2 n$
5	c_5	$\log_2 n$

$$T(n) = (c_1 + c_2) \cdot 1 + (c_3 + c_4 + c_5) \cdot \log_2 n$$

Aufgabe 19

```
1 i = n
2 s = 0
3 while i > 0:
4     s = s + 1
5     i = i // 2
```

Zeile	Kosten	Anzahl
1	c_1	1
2	c_2	1
3	c_3	$\log_2 n$
4	c_4	$\log_2 n$
5	c_5	$\log_2 n$

$$T(n) = (c_1 + c_2) \cdot 1 + (c_3 + c_4 + c_5) \cdot \log_2 n \in \mathcal{O}(\log_2 n)$$

Aufgabe 20

Zu welcher Komplexitätsklasse gehören die folgenden Algorithmen?

(a) Ein Element in einer unsortierten Liste suchen.

(b) Zwei Matrizen multiplizieren.

(c) Eine Liste mit Bubblesort sortieren.

(d) Die Brute-Force-Lösung des Travelling Salesman-Problems.

(e) Eine Liste mit zufällig angeordneten Elementen mit Quicksort sortieren.

Aufgabe 20

(a) Nach einem Element in einer sortierten Liste suchen.

Aufgabe 20

(a) Nach einem Element in einer sortierten Liste suchen.
$\mathcal{O}(\log_2 n)$

Aufgabe 20

(a) Nach einem Element in einer sortierten Liste suchen.
$\mathcal{O}(\log_2 n)$

(b) Zwei Matrizen multiplizieren.

Aufgabe 20

(a) Nach einem Element in einer sortierten Liste suchen.
$\mathcal{O}(\log_2 n)$

(b) Zwei Matrizen multiplizieren.
$\mathcal{O}(n^3)$

Aufgabe 20

(a) Nach einem Element in einer sortierten Liste suchen.
$\mathcal{O}(\log_2 n)$

(b) Zwei Matrizen multiplizieren.
$\mathcal{O}(n^3)$

(c) Eine Liste von Zahlen mit Bubblesort sortieren.

Aufgabe 20

(a) Nach einem Element in einer sortierten Liste suchen.
$\mathcal{O}(\log_2 n)$

(b) Zwei Matrizen multiplizieren.
$\mathcal{O}(n^3)$

(c) Eine Liste von Zahlen mit Bubblesort sortieren.
$\mathcal{O}(n^2)$

Aufgabe 20

(a) Nach einem Element in einer sortierten Liste suchen.
$\mathcal{O}(\log_2 n)$

(b) Zwei Matrizen multiplizieren.
$\mathcal{O}(n^3)$

(c) Eine Liste von Zahlen mit Bubblesort sortieren.
$\mathcal{O}(n^2)$

(d) Die Brute-Force-Lösung des Travelling Salesman-Problems.

Aufgabe 20

(a) Nach einem Element in einer sortierten Liste suchen.
$\mathcal{O}(\log_2 n)$

(b) Zwei Matrizen multiplizieren.
$\mathcal{O}(n^3)$

(c) Eine Liste von Zahlen mit Bubblesort sortieren.
$\mathcal{O}(n^2)$

(d) Die Brute-Force-Lösung des Travelling Salesman-Problems.
$\mathcal{O}(n!)$

Aufgabe 20

(a) Nach einem Element in einer sortierten Liste suchen.
$\mathcal{O}(\log_2 n)$

(b) Zwei Matrizen multiplizieren.
$\mathcal{O}(n^3)$

(c) Eine Liste von Zahlen mit Bubblesort sortieren.
$\mathcal{O}(n^2)$

(d) Die Brute-Force-Lösung des Travelling Salesman-Problems.
$\mathcal{O}(n!)$

(e) Eine Liste von Zufallszahlen mit Quicksort sortieren.

Aufgabe 20

(a) Nach einem Element in einer sortierten Liste suchen.
$\mathcal{O}(\log_2 n)$

(b) Zwei Matrizen multiplizieren.
$\mathcal{O}(n^3)$

(c) Eine Liste von Zahlen mit Bubblesort sortieren.
$\mathcal{O}(n^2)$

(d) Die Brute-Force-Lösung des Travelling Salesman-Problems.
$\mathcal{O}(n!)$

(e) Eine Liste von Zufallszahlen mit Quicksort sortieren.
$\mathcal{O}(n \log_2 n)$