

Aufgabe 1

Beschreibe kurz und präzise (aber ohne mathematische Strenge), was ein Algorithmus ist.

Aufgabe 2

Beschreibe möglichst konkret den Begriff der Laufzeitkomplexität eines Algorithmus und gib wichtige Laufzeitklassen in der Big-Oh-Notation an.

Aufgabe 3

Welche Arten von Algorithmen sind für grosse Problemgrößen nicht mehr handhabbar?

Aufgabe 4

In welcher (minimalen) Laufzeitklasse liegt ein Algorithmus mit der Laufzeit $T(n)$, wenn n die Problemgröße bezeichnet?

(a) $T(n) = 3n + 4n^2 + 2$

(b) $T(n) = (3n + 2)(5n^2 + 1)$

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

(d) $T(n) = \log_2(4n^3)$

Aufgabe 5

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

Aufgabe 6

Eine Implementation eines Algorithmus' hat eine Laufzeitkomplexität von $\mathcal{O}(2^n)$ und benötigt etwa 3s für das Lösen eines Problems der Grösse $n = 10$. Bestimme die ungefähre Laufzeit für ein Problem der Grösse $n = 12$.

Aufgabe 7

Eine Implementation eines Algorithmus' hat eine Laufzeitkomplexität von $\mathcal{O}(\log_2(n))$ und benötigt etwa 3ms für das Lösen eines Problems der Grösse $n = 100$. Bestimme die ungefähre Laufzeit für ein Problem der Grösse $n = 10\,000$.

Aufgabe 8

Analysiere die Laufzeitkomplexität der folgenden Python-Funktion in Abhängigkeit der Problemgrösse n .

```
1 def funktion(n):  
2     s = 0  
3     for i in range(0, n):  
4         s = s + i  
5     return s
```

Aufgabe 9

Analysiere die Laufzeitkomplexität der folgenden Python-Funktion in Abhängigkeit der Problemgrösse n .

```
1 def funktion(n):
2     s = 0
3     for i in range(0, n):
4         for j in range(0, n):
5             s = s + i*j
6     return s
```

Aufgabe 10

Analysiere die Laufzeitkomplexität der folgenden Python-Funktion in Abhängigkeit der Problemgrösse n .

```
1 def funktion(n):
2     s = 0
3     for i in range(0, n):
4         s = s + i
5     for j in range(0, n):
6         s = s + j
7     return s
```

Aufgabe 11

Analysiere die Laufzeitkomplexität der folgenden Python-Funktion in Abhängigkeit der Problemgrösse n .

```
1 def funktion(n):
2     s = 0
3     while n > 0:
4         s = s + n
5         n = n // 2
6     return s
```

Aufgabe 12

Zähle möglichst viele Darstellungsformen von Algorithmen auf und diskutiere ihre Vor- und Nachteile.

Aufgabe 13

Welche Ausgabe macht das Programm für die Eingabe $x = 10$?

| |
|--------------------------|
| Eingabe: x |
| $y \leftarrow 3$ |
| $x \leftarrow x + y$ |
| $y \leftarrow x \bmod 3$ |
| Ausgabe: y |

Aufgabe 14

Welche Ausgabe macht das Programm für die Eingabe $x = 17$?

| | |
|----------------------|----------------------|
| Eingabe: x | |
| $x > 0$ | |
| ja | nein |
| $y \leftarrow x + 1$ | $y \leftarrow x - 1$ |
| Ausgabe: y | |

Aufgabe 15

Welche Ausgabe macht das Programm für die Eingabe $n = 5$?

| |
|----------------------|
| Eingabe: n |
| $n < 10$ |
| $n \leftarrow n + 3$ |
| Ausgabe: n |

Aufgabe 19

Zeige möglichst detailliert, wie die rekursive definierte Funktion den Wert $f(4)$ in der untersten Zeile des folgenden Codefragments berechnet.

```
def f(n):
    if n == 0:
        return 3
    else:
        return 2 + f(n-1)

print(f(4))
```

Aufgabe 20

Zeige möglichst detailliert, wie die rekursive definierte Funktion den Wert $f(5)$ in der untersten Zeile des folgenden Codefragments berechnet.

```
def f(n):
    if n > 0:
        return n + f(n-2)
    else:
        return 7

print(f(5))
```

Aufgabe 21

Eine Person geht mit einer Einkaufsliste in den Supermarkt, sucht die Artikel, legt sie in den Einkaufswagen und streicht sie von der Liste ab. Untersuche kritisch, ob es sich bei dieser Tätigkeit um einen Algorithmus handelt und verwende Fachbegriffe.

Aufgabe 22

Zeige schrittweise, wie der euklidische Algorithmus $\text{ggT}(32, 18)$ berechnet.