Algorithmen (Einführung)

Prüfungsvorbereitung

Aufgabe 1

Ordne die Algorithmen aufgrund ihrerer Laufzeitkomplexität mit dem Symbol $<$ in aufsteigender Reihenfolge.

- Algorithmus A: $O(2^n)$
- Algorithmus B: $O(\log_2 n)$
- Algorithmus C: $O(n^2)$
- Algorithmus D: $O(n!)$
- Algorithmus E: $O(\sqrt{n})$

Aufgabe 1

$B < E < C < A < D$

Da es sich um *asymptotische Laufzeiten*, d. h Laufzeiten für *grosse* n handelt, kann es sein, dass die Dominanz einer der Funktionen erst ab einem ausreichend grossen n erkennbar wird. Zu Illustration dieses Sachverhalts sind hier die Werte der Laufzeitfunktionen für einige n tabelliert:

Aufgabe 2

Vereinfache die Laufzeitkomplexitäten.

(a) $O(2^{n+1})$

(b) $O(\sqrt{4n})$

(c) $O((n^2+n)(n+2)(n+3))$

(d) $O(3)$

Aufgabe 2

(a) $O(2^{n+1}) = O(2 \cdot 2^n) = O(2^n)$

(b) $O(\sqrt{4n}) = 2\sqrt{n} = O(\sqrt{n})$

(c) $O((n^2 + n)(n + 2)(n + 3)) = O(n^4 + \dots) = O(n^4)$

(d) $O(3) = O(1)$

Aufgabe 3

Bestimme die Komplexitätsklasse $O(?)$ der folgenden Python-Funktion in Abhängigkeit der Inputgrösse $n \in \mathbb{N}$.

```python
def funktion(n):
    s = 0
    for i in range(0, n):
        for j in range(0, i):
            s = s + j
    return s
```

Aufgabe 3

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

Zeile	Kosten	Anzahl
2	c_1	1
3	c_2	n
4	c_3	n^2
5	c_4	n^2
6	c_5	1

Aufgabe 3

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

Zeile	Kosten	Anzahl
2	c_1	1
3	c_2	n
4	c_3	n^2
5	c_4	n^2
6	c_5	1

$$T(n) = (c_1 + c_5) \cdot 1 + c_2 \cdot n + (c_3 + c_4) \cdot n^2 \in O(n^2)$$

Aufgabe 4

Bestimme die Komplexitätsklasse $O(?)$ der folgenden Python-Funktion in Abhängigkeit der Inputgrösse $n \in \mathbb{N}$.

```python
def funktion(n):
    s = 0
    for i in range(0, n):
        s = s + i
    for j in range(0, n):
        s = s * j
    return s
```

Aufgabe 4

```
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
```

Zeile	Kosten	Anzahl
2	c_1	1
3	c_2	n
4	c_3	n
5	c_2	n
6	c_4	n
7	c_5	1

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

Aufgabe 5

Ein Programm, das einen Algorithmus mit der Laufzeitkomplexität $O(n^2)$ implementiert, benötigt im schlimmsten Fall 40 Sekunden für die Verarbeitung von 10 000 Objekten vergleichbarer Grösse.

Welche maximale Laufzeit benötigt das Programm auf der gleichen Hardware für die Verarbeitung von 30 000 Objekten vergleichbarer Grösse?

Aufgabe 5

$$\begin{aligned} T(30\,000) &= c \cdot (30\,000)^2 = c \cdot (3 \cdot 10\,000)^2 = 9 \cdot c \cdot 10\,000^2 \\ &= 9 \cdot T(10\,000) = 9 \cdot 40\,\mathrm{s} = 360\,\mathrm{s} = 6\,\mathrm{Min} \end{aligned}$$

Aufgabe 6

Ein Programm, das einen Algorithmus mit der Laufzeitkomplexität $O(\log n)$ implementiert, benötigt im Worst Case 8 Minuten für die Verarbeitung von 10^4 Objekten vergleichbarer Grösse.

Das Programm soll 10^{10} Objekte von vergleichbarer Grösse auf der gleichen Hardware verarbeiten. Schätze die maximale Laufzeit ab.

Aufgabe 6

$$T(10^{10}) = c \cdot \log(10^{10}) = c \cdot \log\left(10^{4 \cdot 2.5}\right) = c \cdot \log\left(\left(10^4\right)^{2.5}\right)$$
$$= 2.5 \cdot c \cdot \log(10^4) = 2.5 \cdot T(10^4) = 2.5 \cdot 8\,\text{min} = 20\,\text{min}$$

Aufgabe 7

Ein Computerprogramm in der Komplexitätsklasse $O(n)$ benötigt für die Verarbeitung von $6 \cdot 10^7$ Objekten im schlimmsten Fall etwa 4 Sekunden.

Welche maximale Laufzeit benötigt das Programm auf der gleichen Hardware für die Verarbeitung von $9 \cdot 10^7$ Objekten vergleichbarer Grösse?

Aufgabe 7

$$
\begin{aligned}
T(9 \cdot 10^7) &= c \cdot 9 \cdot 10^7 = 1.5 \cdot c \cdot 6 \cdot 10^7 = 1.5 \cdot T(6 \cdot 10^7) \\
&= 1.5 \cdot 4\,\mathrm{s} = 6\,\mathrm{s}
\end{aligned}
$$

Das Ergebnis kann natürlich auch direkt mit einem Dreisatz bestimmt werden.

Aufgabe 8

Ein Programm implementiert einen Algorithmus der Komplexitätsklasse $O(n \log n)$. Damit lässt sich ein Problem der Grösse $n = 200$ Worst Case in 5 Millisekunden verarbeiten.

Welche ungefähre maximale Laufzeit benötigt dasselbe Programm auf derselben Hardware für ein Problem der Grösse $n = 8\,000\,000$?

Aufgabe 8

$$
\begin{aligned}
T(8\,000\,000) &= T(200^3) = c \cdot 200^3 \cdot \log_2(200^3) \\
&= c \cdot 200^2 \cdot 200 \cdot 3 \cdot \log(200) \\
&= 40\,000 \cdot 3 \cdot c \cdot 200 \cdot \log(200) \\
&= 120\,000 \cdot T(200) = 120\,000 \cdot 5\,\text{ms} \\
&= 600\,000\,\text{ms} = 10\,\text{min}
\end{aligned}
$$

Aufgabe 9

Ein Programm implementiert einen Algorithmus der Komplexitätsklasse $O(2^n)$. Damit lässt sich ein Problem der Grösse $n = 40$ im schlimmsten Fall in 6 Stunden verarbeiten.

Welche ungefähre maximale Laufzeit benötigt dasselbe Programm auf derselben Hardware für ein Problem der Grösse $n = 45$?

Aufgabe 9

$$\begin{aligned} T(45) &= c \cdot 2^{45} = c \cdot 2^{40} \cdot 2^{5} = 32 \cdot c \cdot 2^{40} \\ &= 32 \cdot T(40) = 32 \cdot 6\,\mathrm{h} = 8 \cdot 24\,\mathrm{h} = 8\,\mathrm{d} \end{aligned}$$

Aufgabe 10

Ein Programm implementiert einen Algorithmus der Komplexitätsklasse $O(n!)$. Damit lässt sich ein Problem der Grösse $n = 7$ im schlimmsten Fall in 10 Sekunden verarbeiten.

Welche ungefähre maximale Laufzeit benötigt dasselbe Programm auf derselben Hardware für ein Problem der Grösse $n = 9$?

Aufgabe 10

$$\begin{aligned} T(9) &= c \cdot 9! = c \cdot 9 \cdot 8 \cdot 7! = 72 \cdot c \cdot 7! \\ &= 72 \cdot T(7) = 72 \cdot 10\,\mathrm{s} = 720\,\mathrm{s} = 12\,\mathrm{min} \end{aligned}$$