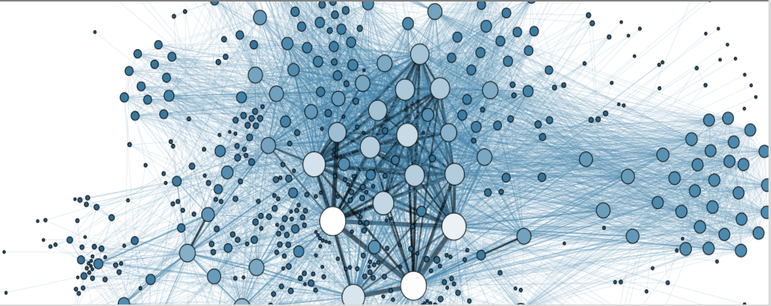
Grundbegriffe der Informatik
Tutorium 38

O-Kalkül und Mastertheorem

Patrick Fetzer, `uxkln@student.kit.edu` | 17.01.2018



Rückblick

Patrick Fetzer,
uxkln@student.kit.edu

Komplexitätstheorie

Mastertheorem

Rückblick

Patrick Fetzer,
uxkln@student.kit.edu

Karlsruher Institut für Technologie

- Was ist $\Omega(f), \Theta(f), \mathcal{O}(f)$?

Patrick Fetzer,
uxkln@student.kit.edu

Rückblick

- Was ist $\Omega(f), \Theta(f), \mathcal{O}(f)$?
- Wieso messen wir nicht einfach Laufzeit in “Anzahl Operationen”?

Obere und untere Schranke

Patrick Fetzer,
uxkln@student.kit.edu

Obere Schranke (Worst-Case Approximation)

$$O(f) = \{g | \exists c \in \mathbb{R}_+ : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : g(n) \leq c \cdot f(n)\}$$

Obere und untere Schranke

Patrick Fetzer,
uxkln@student.kit.edu

Obere Schranke (Worst-Case Approximation)

$$O(f) = \{g | \exists c \in \mathbb{R}_+ : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : g(n) \leq c \cdot f(n)\}$$

Untere Schranke (Best-Case Approximation)

$$\Omega(f) = \{g | \exists c \in \mathbb{R}_+ : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : g(n) \geq c \cdot f(n)\}$$

Obere und untere Schranke

Patrick Fetzer,
uxkln@student.kit.edu

Obere Schranke (Worst-Case Approximation)

$$O(f) = \{g | \exists c \in \mathbb{R}_+ : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : g(n) \leq c \cdot f(n)\}$$

Untere Schranke (Best-Case Approximation)

$$\Omega(f) = \{g | \exists c \in \mathbb{R}_+ : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : g(n) \geq c \cdot f(n)\}$$

Average-Case Approximation

$$\Theta(f) = \{g | \exists c, c' \in \mathbb{R}_+ : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : c \cdot f(n) \leq g(n) \leq c' \cdot f(n)\}$$

Obere und untere Schranke

Patrick Fetzer,
uxkln@student.kit.edu

Obere Schranke (Worst-Case Approximation)

$$O(f) = \{g \mid \exists c \in \mathbb{R}_+ : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : g(n) \leq c \cdot f(n)\}$$

Untere Schranke (Best-Case Approximation)

$$\Omega(f) = \{g \mid \exists c \in \mathbb{R}_+ : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : g(n) \geq c \cdot f(n)\}$$

Average-Case Approximation

$$\Theta(f) = \{g \mid \exists c, c' \in \mathbb{R}_+ : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : c \cdot f(n) \leq g(n) \leq c' \cdot f(n)\}$$

Wiederholung

Patrick Fetzer,
uxkln@student.kit.edu

Ist die Aussage wahr?

Wiederholung

Karlsruher Institut für Technologie

Patrick Fetzer,
uxkln@student.kit.edu

Ist die Aussage wahr?

- $\pi n^{10} \in O(\frac{1}{e^{10}} n^{10})$

Patrick Fetzer,
uxkln@student.kit.edu

Wiederholung

Ist die Aussage wahr?

- $\pi n^{10} \in O(\frac{1}{e^{10}} n^{10})$ Ja

Wiederholung

Patrick Fetzer,
uxkln@student.kit.edu

Karlsruher Institut für Technologie

Ist die Aussage wahr?

- $\pi n^{10} \in O(\frac{1}{e^{10}} n^{10})$ Ja
- $5n^2 + 3 \in O(\frac{1}{2} n^2)$

Patrick Fetzer,
uxkln@student.kit.edu

Wiederholung

Ist die Aussage wahr?

- $\pi n^{10} \in O(\frac{1}{e^{10}} n^{10})$ Ja
- $5n^2 + 3 \in O(\frac{1}{2} n^2)$ Ja

Patrick Fetzer,
uxkln@student.kit.edu

Wiederholung

Ist die Aussage wahr?

- $\pi n^{10} \in O(\frac{1}{e^{10}} n^{10})$ Ja
- $5n^2 + 3 \in O(\frac{1}{2} n^2)$ Ja
- $5^n \in O(3^n)$

Wiederholung

Patrick Fetzer,
uxkln@student.kit.edu

Ist die Aussage wahr?

- $\pi n^{10} \in O(\frac{1}{e^{10}} n^{10})$ Ja
- $5n^2 + 3 \in O(\frac{1}{2} n^2)$ Ja
- $5^n \in O(3^n)$ Nein

Wiederholung

Patrick Fetzer,
uxkln@student.kit.edu

Ist die Aussage wahr?

- $\pi n^{10} \in O(\frac{1}{e^{10}} n^{10})$ Ja
- $5n^2 + 3 \in O(\frac{1}{2} n^2)$ Ja
- $5^n \in O(3^n)$ Nein
- $\log_3(n^2) \in O(7 \log_2(n))$

Wiederholung

Patrick Fetzer,
uxkln@student.kit.edu

Ist die Aussage wahr?

- $\pi n^{10} \in O(\frac{1}{e^{10}} n^{10})$ Ja
- $5n^2 + 3 \in O(\frac{1}{2} n^2)$ Ja
- $5^n \in O(3^n)$ Nein
- $\log_3(n^2) \in O(7 \log_2(n))$ Ja

Patrick Fetzer,
uxkln@student.kit.edu

Wiederholung

Ist die Aussage wahr?

- $\pi n^{10} \in O(\frac{1}{e^{10}} n^{10})$ Ja
- $5n^2 + 3 \in O(\frac{1}{2} n^2)$ Ja
- $5^n \in O(3^n)$ Nein
- $\log_3(n^2) \in O(7 \log_2(n))$ Ja
- $n^n \in O(n!)$

Patrick Fetzer,
uxkln@student.kit.edu

Wiederholung

Ist die Aussage wahr?

- $\pi n^{10} \in O(\frac{1}{e^{10}} n^{10})$ Ja
- $5n^2 + 3 \in O(\frac{1}{2} n^2)$ Ja
- $5^n \in O(3^n)$ Nein
- $\log_3(n^2) \in O(7 \log_2(n))$ Ja
- $n^n \in O(n!)$ Nein

Aufgabe

Patrick Fetzer,
uxkln@student.kit.edu

Zeige, dass $f(n) \in \Theta(g(n))$ mit $f(n) = 2n^4 + 4n^3$ und $g(n) = 5n^4 - 2n^2$

Patrick Fetzer,
uxkln@student.kit.edu


```
r ← 0
for i ← 0 to n/2 do
    s ← 0
    for j ← i to n – i do
        s ← s + j
    od
    r ← s + n * i
    r ← r + s
od
```

Patrick Fetzer,
uxkln@student.kit.edu


```
r ← 0
for i ← 0 to n/2 do
    s ← 0
    for j ← i to n – i do
        s ← s + j
    od
    r ← s + n * i
    r ← r + s
od
```

- Wie oft wird die innere Schleife durchlaufen?

Patrick Fetzer,
uxkln@student.kit.edu


```
r ← 0
for i ← 0 to n/2 do
  s ← 0
  for j ← i to n – i do
    s ← s + j
  od
  r ← s + n * i
  r ← r + s
od
```

- Wie oft wird die innere Schleife durchlaufen? $n - 2i + 1$ mal.

Patrick Fetzer,
uxkln@student.kit.edu


```
r ← 0
for i ← 0 to n/2 do
  s ← 0
  for j ← i to n – i do
    s ← s + j
  od
  r ← s + n * i
  r ← r + s
od
```

- Wie oft wird die innere Schleife durchlaufen? $n - 2i + 1$ mal.
- Wie kommen wir jetzt auf die Gesamtlaufzeit?

Patrick Fetzer,
uxkln@student.kit.edu


```
r ← 0
for i ← 0 to n/2 do
  s ← 0
  for j ← i to n – i do
    s ← s + j
  od
  r ← s + n * i
  r ← r + s
od
```

- Wie oft wird die innere Schleife durchlaufen? $n - 2i + 1$ mal.
- Wie kommen wir jetzt auf die Gesamtlaufzeit?
- $\sum_{i=0}^{n/2} (n - 2i + 1)$

Patrick Fetzer,
uxkln@student.kit.edu


```
r ← 0
for i ← 0 to n/2 do
   s ← 0
   for j ← i to n – i do
      s ← s + j
   od
   r ← s + n * i
   r ← r + s
od
```

- Wie oft wird die innere Schleife durchlaufen? $n - 2i + 1$ mal.
- Wie kommen wir jetzt auf die Gesamtlaufzeit?
- $\sum_{i=0}^{n/2} (n - 2i + 1) = \frac{n}{2}n - 2\sum_{i=0}^{n/2} i + \frac{n}{2}$

Patrick Fetzer,
uxkln@student.kit.edu


```
r ← 0
for i ← 0 to n/2 do
  s ← 0
  for j ← i to n – i do
    s ← s + j
  od
  r ← s + n * i
  r ← r + s
od
```

- Wie oft wird die innere Schleife durchlaufen? $n - 2i + 1$ mal.
- Wie kommen wir jetzt auf die Gesamtlaufzeit?
- $\sum_{i=0}^{n/2}(n - 2i + 1) = \frac{n}{2}n - 2\sum_{i=0}^{n/2} i + \frac{n}{2} = \frac{n^2}{2} + \frac{n}{2} - 2\frac{\frac{n}{2}\cdot\left(\frac{n}{2}+1\right)}{2}$

Patrick Fetzer,
uxkln@student.kit.edu


```
r ← 0
for i ← 0 to n/2 do
  s ← 0
  for j ← i to n – i do
    s ← s + j
  od
  r ← s + n * i
  r ← r + s
od
```

- Wie oft wird die innere Schleife durchlaufen? $n - 2i + 1$ mal.
- Wie kommen wir jetzt auf die Gesamtlaufzeit?
- $\sum_{i=0}^{n/2}(n - 2i + 1) = \frac{n}{2}n - 2\sum_{i=0}^{n/2} i + \frac{n}{2} = \frac{n^2}{2} + \frac{n}{2} - 2\frac{\frac{n}{2}\cdot\left(\frac{n}{2}+1\right)}{2} = \frac{n^2}{2} + \frac{n}{2} - \frac{n^2}{4} - \frac{n}{2}$

```
r ← 0
for i ← 0 to n/2 do
  s ← 0
  for j ← i to n – i do
    s ← s + j
  od
  r ← s + n * i
  r ← r + s
od
```

- Wie oft wird die innere Schleife durchlaufen? $n - 2i + 1$ mal.
- Wie kommen wir jetzt auf die Gesamtlaufzeit?
- $\sum_{i=0}^{n/2}(n-2i+1) = \frac{n}{2}n - 2\sum_{i=0}^{n/2} i + \frac{n}{2} = \frac{n^2}{2} + \frac{n}{2} - 2\frac{\frac{n}{2}\cdot\left(\frac{n}{2}+1\right)}{2} = \frac{n^2}{2} + \frac{n}{2} - \frac{n^2}{4} - \frac{n}{2} = \frac{1}{4}n^2$

Patrick Fetzer,
uxkln@student.kit.edu

Mastertheorem

Karlsruher Institut für Technologie

Formel für Mastertheorem

Rekursive Komplexitätsformeln der Form

$$T(n) = a \cdot T(\tfrac{n}{b}) + f(n)$$

lassen sich mit dem Mastertheorem Komplexitätsklassen zuordnen.

Auflösung des Mastertheorem

Fall 1: Wenn $f \in \mathcal{O}(n^{\log_b a - \varepsilon})$ für ein $\varepsilon > 0$ ist, dann ist $T \in \Theta(n^{\log_b a})$.

Fall 2: Wenn $f \in \Theta(n^{\log_b a})$ ist, dann ist $T \in \Theta(n^{\log_b a} \log n)$.

Fall 3: Wenn $f \in \Omega(n^{\log_b a + \varepsilon})$ für ein $\varepsilon > 0$ ist, und wenn es eine Konstante d gibt mit $0 < d < 1$, so dass für alle hinreichend großen n gilt $af(n/b) \leq df$, dann ist $T \in \Theta(f)$.

Aufgaben zum Mastertheorem

Patrick Fetzer,
uxkln@student.kit.edu

Aufgaben zum Mastertheorem

Patrick Fetzer,
uxkln@student.kit.edu

- $T(n) := 2T(\frac{n}{4}) + \sqrt{n}$

Aufgaben zum Mastertheorem

Patrick Fetzer,
uxkln@student.kit.edu

- $T(n) := 2T(\frac{n}{4}) + \sqrt{n}$, also $a = 2, b = 4, f(n) = \sqrt{n}$

Aufgaben zum Mastertheorem

Patrick Fetzer,
uxkln@student.kit.edu

- $T(n) := 2T(\frac{n}{4}) + \sqrt{n}$, also $a = 2$, $b = 4$, $f(n) = \sqrt{n}$, also zweiter Fall des Mastertheorems

Aufgaben zum Mastertheorem

Karlsruher Institut für Technologie

Patrick Fetzer,
uxkln@student.kit.edu

- $T(n) := 2T(\frac{n}{4}) + \sqrt{n}$, also $a = 2, b = 4, f(n) = \sqrt{n}$, also zweiter Fall des Mastertheorems. $T \in \Theta(\sqrt{n}\log n)$

Patrick Fetzer,
uxkln@student.kit.edu

Aufgaben zum Mastertheorem

Karlsruher Institut für Technologie

- $T(n) := 2T(\frac{n}{4}) + \sqrt{n}$, also $a = 2, b = 4, f(n) = \sqrt{n}$, also zweiter Fall des Mastertheorems. $T \in \Theta(\sqrt{n} \log n)$
- $T(n) := 3T(\frac{n}{2}) + n \log n$

Aufgaben zum Mastertheorem

Patrick Fetzer,
uxkln@student.kit.edu

- $T(n) := 2T(\frac{n}{4}) + \sqrt{n}$, also $a = 2, b = 4, f(n) = \sqrt{n}$, also zweiter Fall des Mastertheorems. $T \in \Theta(\sqrt{n} \log n)$
- $T(n) := 3T(\frac{n}{2}) + n \log n$, also $a = 3, b = 2, f(n) = n \log n$

Aufgaben zum Mastertheorem

Patrick Fetzer,
uxkln@student.kit.edu

- $T(n) := 2T(\frac{n}{4}) + \sqrt{n}$, also $a = 2, b = 4, f(n) = \sqrt{n}$, also zweiter Fall des Mastertheorems. $T \in \Theta(\sqrt{n}\log n)$
- $T(n) := 3T(\frac{n}{2}) + n\log n$, also $a = 3, b = 2, f(n) = n\log n$, also erster Fall des Mastertheorems

Aufgaben zum Mastertheorem

Patrick Fetzer,
uxkln@student.kit.edu

- $T(n) := 2T(\frac{n}{4}) + \sqrt{n}$, also $a = 2$, $b = 4$, $f(n) = \sqrt{n}$, also zweiter Fall des Mastertheorems. $T \in \Theta(\sqrt{n}\log n)$
- $T(n) := 3T(\frac{n}{2}) + n\log n$, also $a = 3$, $b = 2$, $f(n) = n\log n$, also erster Fall des Mastertheorems, $T \in \Theta(n^{\log_2 3})$

Patrick Fetzer,
uxkln@student.kit.edu

Aufgaben zum Mastertheorem

Karlsruher Institut für Technologie

- $T(n) := 2T(\frac{n}{4}) + \sqrt{n}$, also $a = 2$, $b = 4$, $f(n) = \sqrt{n}$, also zweiter Fall des Mastertheorems. $T \in \Theta(\sqrt{n}\log n)$
- $T(n) := 3T(\frac{n}{2}) + n\log n$, also $a = 3$, $b = 2$, $f(n) = n\log n$, also erster Fall des Mastertheorems, $T \in \Theta(n^{\log_2 3})$
- $T(n) := 4T(\frac{n}{2}) + n^2\sqrt{n}$

Aufgaben zum Mastertheorem

Patrick Fetzer,
uxkln@student.kit.edu

- $T(n) := 2T(\frac{n}{4}) + \sqrt{n}$, also $a = 2, b = 4, f(n) = \sqrt{n}$, also zweiter Fall des Mastertheorems. $T \in \Theta(\sqrt{n}\log n)$
- $T(n) := 3T(\frac{n}{2}) + n\log n$, also $a = 3, b = 2, f(n) = n\log n$, also erster Fall des Mastertheorems, $T \in \Theta(n^{\log_2 3})$
- $T(n) := 4T(\frac{n}{2}) + n^2\sqrt{n}$, also $a = 4, b = 2, f(n) = n^2\sqrt{n}$

Patrick Fetzer,
uxkln@student.kit.edu

Aufgaben zum Mastertheorem

Karlsruher Institut für Technologie

- $T(n) := 2T(\frac{n}{4}) + \sqrt{n}$, also $a = 2$, $b = 4$, $f(n) = \sqrt{n}$, also zweiter Fall des Mastertheorems. $T \in \Theta(\sqrt{n}\log n)$
- $T(n) := 3T(\frac{n}{2}) + n\log n$, also $a = 3$, $b = 2$, $f(n) = n\log n$, also erster Fall des Mastertheorems, $T \in \Theta(n^{\log_2 3})$
- $T(n) := 4T(\frac{n}{2}) + n^2\sqrt{n}$, also $a = 4$, $b = 2$, $f(n) = n^2\sqrt{n}$, also dritter Fall des Mastertheorems

Aufgaben zum Mastertheorem

Patrick Fetzer,
uxkln@student.kit.edu

- $T(n) := 2T(\frac{n}{4}) + \sqrt{n}$, also $a = 2$, $b = 4$, $f(n) = \sqrt{n}$, also zweiter Fall des Mastertheorems. $T \in \Theta(\sqrt{n} \log n)$
- $T(n) := 3T(\frac{n}{2}) + n \log n$, also $a = 3$, $b = 2$, $f(n) = n \log n$, also erster Fall des Mastertheorems, $T \in \Theta(n^{\log_2 3})$
- $T(n) := 4T(\frac{n}{2}) + n^2\sqrt{n}$, also $a = 4$, $b = 2$, $f(n) = n^2\sqrt{n}$, also dritter Fall des Mastertheorems, $T \in \Theta(n^2\sqrt{n})$.

Aufgaben zum Mastertheorem

Patrick Fetzer,
uxkln@student.kit.edu

- $T(n) = 2T(\frac{n}{2}) + 10n$
- $T(n) = 20n^2 + 8T(\frac{n}{2})$
- $T(n) = 4T(\frac{n}{4}) + n^2$

Grundbegriffe der Informatik

Patrick Fetzer,
`uxkln@student.kit.edu`