Grundbegriffe der Informatik
WS 2011/12
Tutorium in der Woche 9
Gehalten in den Tutorien Nr. 10, Nr. 14

Philipp Basler (philippbasler@googlemail.com)
Nils Braun (area51.nils@googlemail.com)

KIT - Karlsruher Institut für Technologie

19.12.2011 & 12.12.2011

Inhaltsverzeichnis

1 Übungsblätter

2 Groß-O-Notation

3 Laufzeiten

4 Schluss

1 **Übungsblätter**

2 Groß-O-Notation

3 Laufzeiten

4 Schluss

Informationen zum nächsten Blatt

Blatt Nr. 9

Abgabetermin	23.12.11
Abgabeort	Briefkasten
Themen	Laufzeiten und O,Θ,Ω
Maximale Punkte	18

Häufige Fehler auf dem letzten Übungsblatt

Blatt Nr. 8

- 1. Aufgabe: alles ok
- 2. Aufgabe: Isormophismus als Funktion -> Schreibt die Quell- und Zielmenge dazu, z.b. $\Phi : G_0 \rightarrow G_1$. Weiterhin ist ein Isomorphismus nicht äquivalent zu einem isomorphen Graph.
- 3. Aufgabe: Unterschied vollständig und unvollständiger Binärbaum
- 4. Aufgabe: Auch der 0-Pfad ist ein Pfad!

Was bleibt?

Was ist eine Äquivalenzrelation?

Was bleibt?

Was ist eine Äquivalenzrelation?

Eine symmetrische, transitive und reflektive Relation!

Was bleibt?

Was ist die schnellste Methode um die Wegematrix zu berechnen?

Was bleibt?

Was ist die schnellste Methode um die Wegematrix zu berechnen?

Warshall-Agorithmus

Was bleibt?

Wie geht der erste Schritt des Warshall-Algorithmus?

Was bleibt?

Wie geht der erste Schritt des Warshall-Algorithmus?

Die Adjazenzmatrix abschreiben und in die Diagonale Einsen auffüllen.

1 Übungsblätter

2 Groß-O-Notation

3 Laufzeiten

4 Schluss

Laufzeiten

Wir hatten uns Laufzeiten angeschaut.

Laufzeiten

Wir hatten uns Laufzeiten angeschaut. Aber mit was wollen wir die messen?

Abschätzung

Wir wollen mit n die Problemgröße bezeichnen.

Abschätzung

Wir wollen mit n die Problemgröße bezeichnen.
Welche Abschätzungen sollen wir dann machen?

Asymptotisches Wachstum

Definition

Zwei Funktionen $f, g : \mathbb{N}_0 \to \mathbb{R}_0^+$ wachsen asymptotisch genauso schnell, wenn es zwei Konstanten $c, c' \in \mathbb{R}^+$ gibt, so dass gilt

$$\exists n' \in \mathbb{N}_0 \; \forall n > n' : \; cf(n) \leq g(n) \leq c' f(n)$$

Man schreibt dann auch

$$f \asymp g$$

Asymptotisches Wachstum

Definition

Zwei Funktionen $f, g : \mathbb{N}_0 \to \mathbb{R}_0^+$ wachsen asymptotisch genauso schnell, wenn es zwei Konstanten $c, c' \in \mathbb{R}^+$ gibt, so dass gilt

$$\exists n' \in \mathbb{N}_0 \; \forall n > n' : \; cf(n) \leq g(n) \leq c' f(n)$$

Man schreibt dann auch

$$f \asymp g$$

Diese Relation ist eine Äquivalenzrelation!

Äquivalenzrelation

Die Relation $\asymp$ ist eine Äquivalenzrelation

symmetrisch Wähle einfach $c = 1/2$ und $c' = 2$

reflexiv Wähle einfach $1/c$ und $1/c'$

transitiv Hässlich aber auch machbar

Äquivalenzklassen

Definition

$\Theta(f)$ ist die Menge aller Funktionen g, die asymptotisch genauso schnell wachsen wie f, also

$$\Theta(f) = \{g \mid f \asymp g\}$$

Äquivalenzklassen

Definition

$\Theta(f)$ ist die Menge aller Funktionen g, die asymptotisch genauso schnell wachsen wie f, also

$$\Theta(f) = \{g \mid f \asymp g\}$$

Wir bauen eine Art Bereich um f, in dem g liegen darf.

Asymptotisches Wachstum - nach oben und unten

Definition

Für zwei Funktionen $f, g : \mathbb{N}_0 \to \mathbb{R}_0^+$ definiert man:

$$g \preceq f \qquad \exists c \in \mathbb{R}^+ \ \exists n' \in \mathbb{N}_0 \ \forall n > n' : \ g(n) \leq cf(n)$$

$$g \succeq f \qquad \exists c \in \mathbb{R}^+ \ \exists n' \in \mathbb{N}_0 \ \forall n > n' : \ g(n) \geq cf(n)$$

Asymptotisches Wachstum - nach oben und unten

Definition

Für zwei Funktionen $f, g : \mathbb{N}_0 \to \mathbb{R}_0^+$ definiert man:

$$g \preceq f \qquad \exists c \in \mathbb{R}^+ \; \exists n' \in \mathbb{N}_0 \; \forall n > n' : \; g(n) \leq cf(n)$$

$$g \succeq f \qquad \exists c \in \mathbb{R}^+ \; \exists n' \in \mathbb{N}_0 \; \forall n > n' : \; g(n) \geq cf(n)$$

Diese Relationen sind keine Äquivalenzrelation!

Äquivalenzklassen

Definition

$O(f)$ ist die Menge aller Funktionen g, die asymptotisch höchstens so schnell wachsen wie f, also

$$O(f) = \{g \mid g \preceq f\}$$

$\Omega(f)$ ist die Menge aller Funktionen g, die asymptotisch mindestens so schnell wachsen wie f, also

$$\Omega(f) = \{g \mid g \succeq f\}$$

Äquivalenzklassen

Definition

$O(f)$ ist die Menge aller Funktionen g, die asymptotisch höchstens so schnell wachsen wie f, also

$$O(f) = \{g \mid g \preceq f\}$$

$\Omega(f)$ ist die Menge aller Funktionen g, die asymptotisch mindestens so schnell wachsen wie f, also

$$\Omega(f) = \{g \mid g \succeq f\}$$

O ist eine Art Abschätzung nach oben. Θ ist die Mindesgrenze nach unten.

Achtung!

Machmal sieht man auch leider das:

$$f = \Theta(g) \qquad h = (n^3) \qquad k = \Omega(f + g)$$

Achtung!

Machmal sieht man auch leider das:

$$f = \Theta(g) \qquad h = (n^3) \qquad k = \Omega(f + g)$$

Da gibt es eigentlich keine Gleichheit!

Grenzwertabschätzung

Wir können das auch anders schreiben:

$$f \in O(g) \iff 0 \leq \limsup_{n\to\infty} \frac{f(n)}{g(n)} < \infty$$

$$f \in \Omega(g) \iff 0 < \liminf_{n\to\infty} \frac{f(n)}{g(n)} \leq \infty$$

$$f \in \Theta(g) \iff 0 < \liminf_{n\to\infty} \frac{f(n)}{g(n)} \leq \limsup_{n\to\infty} \frac{f(n)}{g(n)} < \infty$$

Grenzwertabschätzung

Wir können das auch anders schreiben:

$$f \in O(g) \iff 0 \leq \limsup_{n\to\infty} \frac{f(n)}{g(n)} < \infty$$

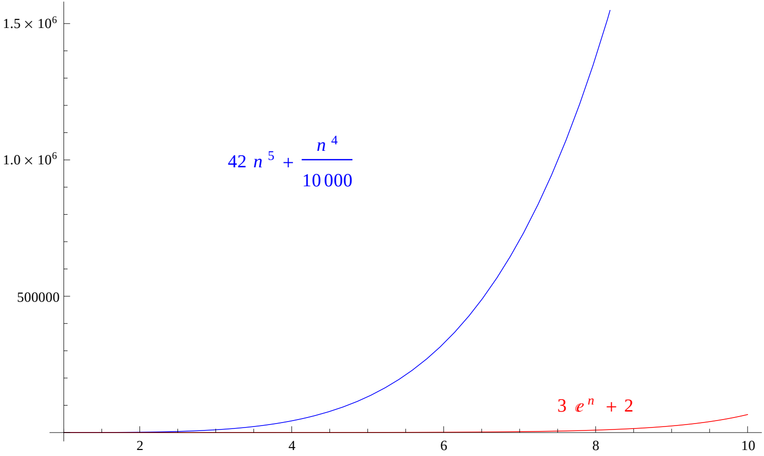
$$f \in \Omega(g) \iff 0 < \liminf_{n\to\infty} \frac{f(n)}{g(n)} \leq \infty$$

$$f \in \Theta(g) \iff 0 < \liminf_{n\to\infty} \frac{f(n)}{g(n)} \leq \limsup_{n\to\infty} \frac{f(n)}{g(n)} < \infty$$

Oftmals existiert sogar lim und wir können lim inf und lim sup vergessen!

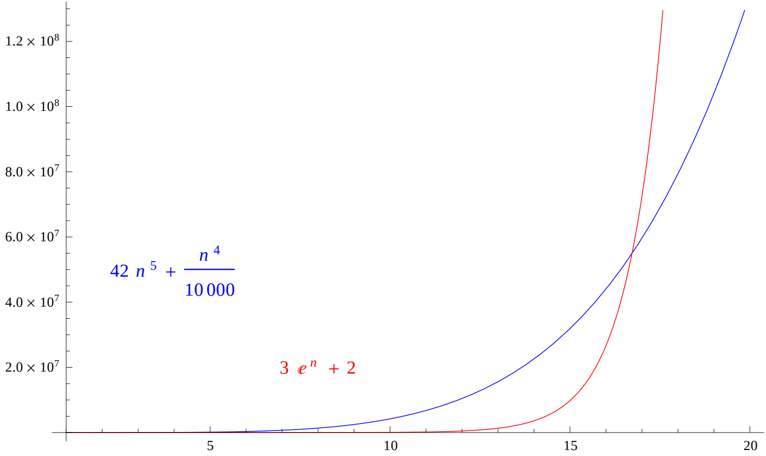
Graphische Beispiele

$$42n^5 + 10^{-5}n^4 \asymp 3e^n + 2$$



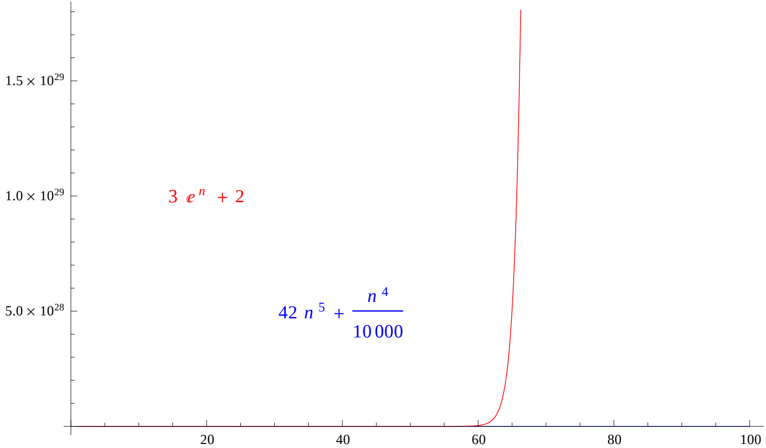
Graphische Beispiele

$$42n^5 + 10^{-5}n^4 \asymp 3e^n + 2$$



Graphische Beispiele

$$42n^5 + 10^{-5}n^4 \asymp 3e^n + 2$$



Polynome

Betrachten wir zwei Polynome $f(n) = n^4 + n^3$ und $g(n) = n^2$:
Der Quotient

$$\frac{f(n)}{g(n)} = \frac{n^4 + n^3}{n^2} = n^2 + n \to \infty$$

Also ist $\lim f/g > 0$ und damit

$$g \in O(f) \qquad f \in \Omega(g)$$

aber $\lim f/g = \infty$ also

$$f \notin O(g) \qquad g \notin \Omega(f)$$

und vor allem

$$f \notin \Theta(g)$$

Polynome

Betrachten wir zwei Polynome $f(n) = n^4 + n^3$ und $g(n) = n^2$:

$g \in O(f)$

Es ist auch für $n > 1$

$$g(n) = n^2 \leq n^3 < 4n^3 = 2(n^3 + n^3) < 2(n^4 + n^3) = 2f(n)$$

Also gibt es ein c in $\mathbb{R}^+$ (nämlich 2), so dass es ein $n_0 \in \mathbb{N}$ gibt (nämlich 1), so dass für alle $n > 1$ gilt:

$$g(n) \leq cf(n)$$

Polynome

Betrachten wir zwei Polynome $f(n) = n^4 + n^3$ und $g(n) = n^2$:

Ann. $f \in O(g)$

Dann müsste es ein $c' \in \mathbb{R}^+$ geben, so dass es ein $n_0 \in \mathbb{N}$ gibt, so dass für alle $n > n_0$ gilt:

$$f(n) < c'g(n)$$

Wähle jetzt ein $n' > n_0$ so, dass $f(n'), g(n') \neq 0$ gilt. Insbesondere wäre dann

$$\frac{f(n')}{g(n')} = n^2 + n \leq c$$

Dies ist ein Widerspruch

Logarithmus

Es ist nach den Logarithmusregeln

$$n = b^{\log_b n} \qquad n = a^{\log_a n}$$

Logarithmus

Es ist nach den Logarithmusregeln

$$n = b^{\log_b n} \qquad n = a^{\log_a n}$$

Aus dem letzen machen wir:

$$a^{\log_a n} = \left(b^{\log_b a}\right)^{\log_a n} = b^{\log_b a \log_a n}$$

Logarithmus

Es ist nach den Logarithmusregeln

$$n = b^{\log_b n} \qquad n = a^{\log_a n}$$

Aus dem letzen machen wir:

$$a^{\log_a n} = \left(b^{\log_b a}\right)^{\log_a n} = b^{\log_b a \log_a n}$$

Wir haben also

$$b^{\log_b n} = b^{\log_b a \log_a n} \qquad \log_b n = \log_b a \log_a n$$

Logarithmus

Es ist nach den Logarithmusregeln

$$n = b^{\log_b n} \qquad n = a^{\log_a n}$$

Aus dem letzen machen wir:

$$a^{\log_a n} = \left(b^{\log_b a}\right)^{\log_a n} = b^{\log_b a \log_a n}$$

Wir haben also

$$b^{\log_b n} = b^{\log_b a \log_a n} \qquad \log_b n = \log_b a \log_a n$$

Setze $c = c' = \log_b a$, dann ist

$$c \log_a n \leq \log_b n \leq c' \log_a n$$

Rechenregeln

Einige Rechenregeln im O-Kalkül

- Für $a > 0$ ist $af \in \Theta(f)$
- Für $0 < a < b$ ist $n^a \preceq n^b$
- Für $a, b > 1$ ist $n^a \preceq b^n$
- Für Polynome f, g gilt:

$$grad\ f = grad\ g \iff f \asymp g$$

- Für $a, b > 0$ gilt $\log_a(n) \in \Theta(\log_b n)$

Rechenregeln

Einige Rechenregeln im O-Kalkül:

- $f \in O(g) \iff g \in \Omega(f)$
- $\Theta(f) = O(f) \cap \Omega(f)$ oder $f \asymp g \iff f \preceq g \wedge f \succeq g$
- $O(f_1) + O(f_2) = O(f_1 + f_2)$
- Wenn $g \in O(f)$, dann ist auch $O(g) \subseteq O(f)$ und $O(f + g) = O(f)$

1. Übungsblätter
2. Groß-O-Notation
3. **Laufzeiten**
4. Schluss

Algorithmus

```
int x = 1;
for( int i = 1; i < n; i++ ) {
 x = a * x;
}
return(x);
```

Algorithmus

```
int x = 1;
for( int i = 1; i < n; i++ ) {
 x = a * x;
}
return(x);
```

Was tut der?

Algorithmus

```
int x = 1;
for( int i = 1; i < n; i++ ) {
 x = a * x;
}
return(x);
```

Was tut der? Welche Laufzeit hat er?

Laufzeiten

```
int x = 1;
```

Laufzeiten

```
int x = 1;
```

$O(1)$

Laufzeiten

```
int x = 1;
```

$O(1)$

```
for( int i = 1; i < n; i++ ) {
```

Laufzeiten

```
int x = 1;
```

$O(1)$

```
for( int i = 1; i < n; i++ ) {
```

$O(1), O(1)$

Laufzeiten

```
int x = 1;
```
$O(1)$
```
for( int i = 1; i < n; i++ ) {
```
$O(1), O(1)$
```
 x = a * x;
```

Laufzeiten

```
int x = 1;
```
$O(1)$
```
for( int i = 1; i < n; i++ ) {
```
$O(1), O(1)$
```
 x = a * x;
```
$O(1)$

Laufzeiten

```
int x = 1;
```
$O(1)$
```
for( int i = 1; i < n; i++ ) {
```
$O(1), O(1)$
```
 x = a * x;
```
$O(1)$
```
}
```

Laufzeiten

```
int x = 1;
```
$O(1)$
```
for( int i = 1; i < n; i++ ) {
```
$O(1), O(1)$
```
 x = a * x;
```
$O(1)$
```
}
```
n-mal

Laufzeiten

`int x = 1;`

$O(1)$

`for( int i = 1; i < n; i++ ) {`

$O(1), O(1)$

` x = a * x;`

$O(1)$

`}`

n-mal

`return(x);`

Laufzeiten

`int x = 1;`

$O(1)$

`for( int i = 1; i < n; i++ ) {`

$O(1), O(1)$

` x = a * x;`

$O(1)$

`}`

n-mal

`return(x);`

$O(1)$

Laufzeiten

`int x = 1;`

$O(1)$

`for( int i = 1; i < n; i++ ) {`

$O(1), O(1)$

` x = a * x;`

$O(1)$

`}`

n-mal

`return(x);`

$O(1)$

Laufzeiten

```
int x = 1;
```
$O(1)$
```
for( int i = 1; i < n; i++ ) {
```
$O(1), O(1)$
```
 x = a * x;
```
$O(1)$
```
}
```
n-mal
```
return(x);
```
$O(1)$

$$O(1 + n + n + 1) = O(n)$$

1 Übungsblätter

2 Groß-O-Notation

3 Laufzeiten

4 Schluss

Was ihr nun wissen solltet

- Was O, Θ und Ω bedeuten
- Wie ihr asymptotisch schneller oder langsamer werdet
- Wie man Laufzeiten von Algorithmen bestimmt

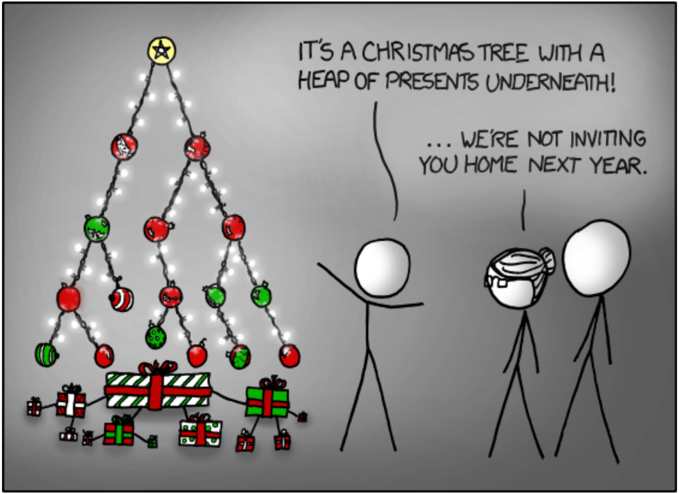


Abbildung: `www.xkcd.com`

Kontakt via E-Mail an Philipp Basler oder Nils Braun
gbi.ugroup.hostzi.com

Frohe Weihnachten und einen guten Rutsch ins neue Jahr!