Grundbegriffe der Informatik

Tutorium 1 - 9. Sitzung

Dennis Felsing

dennis.felsing@student.kit.edu
http://www.stud.uni-karlsruhe.de/~ubcqr/2010w/tut_gbi/

2010-12-20

Überblick

Berechnung der Wegematrix nach Warshall

for $i \leftarrow 0$ **to** $n-1$ **do**
  for $j \leftarrow 0$ **to** $n-1$ **do**

$$W_{ij} \leftarrow \begin{cases} 1 & \text{falls } i = j \\ A_{ij} & \text{falls } i \neq j \end{cases}$$

  od
od
for $k \leftarrow 0$ **to** $n-1$ **do**
  for $i \leftarrow 0$ **to** $n-1$ **do**
    for $j \leftarrow 0$ **to** $n-1$ **do**
      $W_{ij} \leftarrow \max(\, W_{ij}, \min(W_{ik}, W_{kj})\,)$
    od
  od
od

Warshall-Algorithmus

In Worten:

- Schritt 0: Füge Kante (i, j) hinzu, wenn es einen Weg von i nach j über Knoten 0 gibt.
- Schritt 1: Füge Kante (i, j) hinzu, wenn es einen Weg von i nach j über Knoten 1 gibt.
- Fortsetzen bis Knoten $n - 1$

Warshall-Algorithmus

- Was ist $\mathbb{G}_k$? $\{0, 1, 2, ..., k-1\}$
- Was macht der Algorithmus? Wegematrix berechnen.
- Warum stimmt am Ende das Ergebnis? Wie kann man das beweisen? Dazu braucht man eine **Schleifeninvariante**: Nach k Durchläufen der äußeren Schleife ist $W[i, j] = 1$, genau dann wenn es einen wiederholungsfreien Pfad von i nach j gibt, bei dem alle Zwischenknoten in $\mathbb{G}_k$ sind.
- Welche Laufzeit hat der Algorithmus? Erste Schleife n^2, zweite Schleife n^3. Wie können wir sowas richtig aufschreiben?

Asymptotisches Wachstum

Definitionen

Seien $f : \mathbb{N}_0 \to \mathbb{R}_0^+$, $g : \mathbb{N}_0 \to \mathbb{R}_0^+$.

$$
\begin{aligned}
f \asymp g &\Leftrightarrow f \text{ wächst größenordnungsmäßig so schnell wie } g \\
&\Leftrightarrow \exists c, c' \in \mathbb{R}_+ : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : cf(n) \leq \\
&\quad g(n) \leq c' f(n) \\
\Theta(f) &= \{g \mid f \asymp g\}
\end{aligned}
$$

Beispiel Polynome

$f(n) = 42n^6 - 11n^3 + 111n^2 - 3$
$g(n) = 66n^6 + 130n^5$
Zu zeigen: $f \asymp g$

$$\begin{aligned} 1 \cdot f(n) &= 42n^6 - 11n^3 + 111n^2 - 3 \\ &\leq 42n^6 + 111n^2 \\ &\leq 42n^6 + 111n^5 \\ &\leq 66n^6 + 130n^5 \\ &= g(n) \end{aligned}$$

Beispiel Polynome

$f(n) = 42n^6 - 11n^3 + 111n^2 - 3$
$g(n) = 66n^6 + 130n^5$
Zu zeigen: $f \asymp g$

$$
\begin{aligned}
g(n) &= 66n^6 + 130n^5 \\
&\leq 66n^6 + 130n^6 \\
&= 196n^6 \\
&= 294n^6 - 77n^6 - 21n^6 \\
&\leq 294n^6 - 77n^3 - 21 \\
&\leq 294n^6 - 77n^3 - 21 + 777n^2 \\
&= 7 \cdot (42n^6 - 11n^3 + 111n^2 - 3) \\
&= 7 \cdot f(n)
\end{aligned}
$$

Beispiel Polynome

$f(n) = 42n^6 - 11n^3 + 111n^2 - 3$
$g(n) = 66n^6 + 130n^5$
Zu zeigen: $f \asymp g$

$\Rightarrow 1 \cdot f(n) \leq g(n) \leq 7 \cdot f(n)$
$\Rightarrow f \asymp g$
Solche Umformungen funktionieren immer, wenn f und g Polynome gleichen Grades sind. Daher gilt für konstante $a, b \in \mathbb{R}_+$: $\Theta(a \cdot f(n)) = \Theta(b \cdot f(n))$.

Beispiel Logarithmen

Behauptung

$log_2(n) \in \Theta(log_8(n))$

Hinweis

$\forall a \in \mathbb{R}_+, n \in \mathbb{N}_+ : a^{log_a(n)} = n$

Allgemein Logarithmen

Behauptung

$log_b(n) \in \Theta(log_a(n))$

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

$\Rightarrow log_b n = log_b a \cdot log_a n$

Setze $c = c' = log_b a$, dann gilt:

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

Da die Basis also egal ist, kann man einfach $\Theta(logn)$ schreiben.

Asymptotisches Wachstum

$\asymp$ ist eine Äquivalenzrelation

- Was bedeutet das? $\asymp$ ist reflexiv, transitiv und symmetrisch.
- Was bedeutet Reflexivität in diesem Fall? Jede Funktion wächst größenordnungsmäßig so schnell wie sie selbst.
- Was bedeutet Symmetrie in diesem Fall? $f \asymp g \Leftrightarrow g \asymp f$
- Was bedeutet Transitivität in diesem Fall?
  $f \asymp g \wedge g \asymp h \Rightarrow f \asymp h$

Obere und untere Schranken

Definitionen

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

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

$$\Theta(f) = O(f) \cap \Omega(f)$$

In Worten

$O(f)$: Menge der Funktionen, die asymptotisch höchstens so schnell wachsen wie f.

$\Omega(f)$: Menge der Funktionen, die asymptotisch mindestens so schnell wachsen wie f.

Beispiel 1

Hinweis

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

$f(n) = 2n^4$
$g(n) = 3n^3$
Behauptung: $g \in O(f)$

Wähle $c = 1, n_0 = 2$.Es gilt $\forall n \geq n_0 : g(n) \leq c \cdot f(n)$. Somit $g \in O(f)$.

Beispiel 2

Hinweis

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

$f(n) = logn$
$g(n) = n^2$
Behauptung: $g \notin O(f)$

Annahme: $g \in O(f)$. Dann gibt es ein c, so dass für alle n ab einem n_0 gilt: $n^2 \leq c \cdot logn$.Da $logn \leq n$, muss auch gelten $n \cdot logn \leq c \cdot logn$. Somit ist $c \geq n.n$ kann aber beliebig groß werden. Es gibt also kein konstantes c, so dass $\forall n \geq n_0 : c \geq n$. ↯
Somit ist $g \notin O(f)$. □

Beispiel 3

Hinweis

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

$f(n) = 42n^6 - 11n^3 + 111n^2 - 3$
$g(n) = 66n^6 + 130n^5$
Behauptung: $g \in \Omega(f)$

Wähle $c = 1, n_0 = 0. \forall n \geq n_0 : g(n) \geq f(n)$. Somit $g \in \Omega(f)$.
Eigentlich klar, denn wir haben vorhin schon gezeigt, dass f und g asymptotisch gleich schnell wachsen, dann muss g auch mindestens so schnell wachsen wie f.

Rechenregeln

Wir definieren für Mengen
$M_1 + M_2 = \{g_1 + g_2 \mid g_1 \in M_1 \wedge g_2 \in M_2\}$.

$$O(f_1) + O(f_2) = O(f_1 + f_2)$$

Algorithmus analysieren

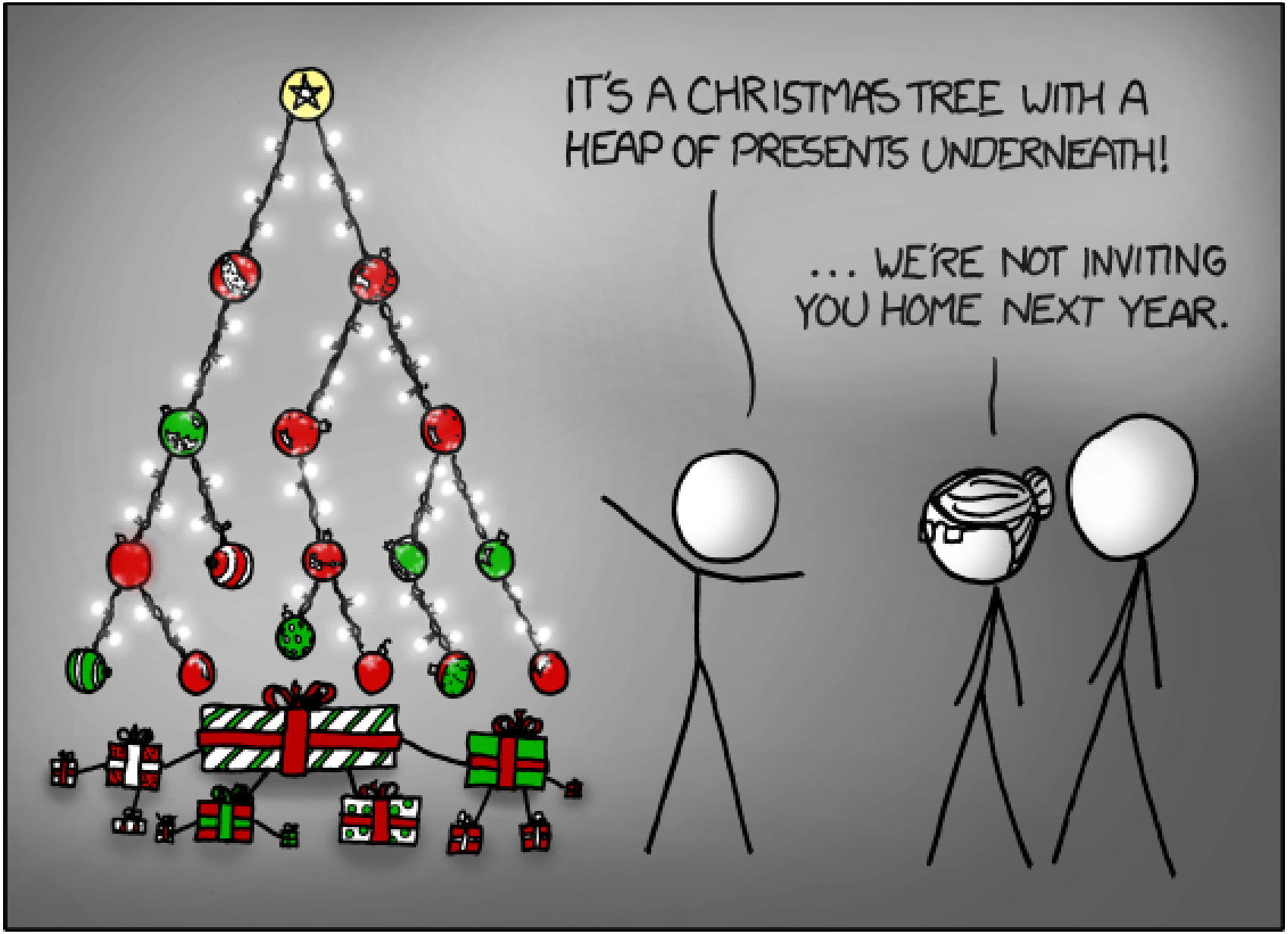
```
for k ← 1 to n do
  for i ← 1 to n/2 do
    for j ← 1 to n/3 do
      print a[j-1]
    od
  od
od
```

Asymptotischer Aufwand

$n \cdot \frac{n}{2} \cdot \frac{n}{3} = \frac{1}{6} \cdot n^3 \in \Theta(n^3)$

Überblick

1. **Warshall-Algorithmus**
   - Algorithmus
   - Verständnis
2. **Groß-O-Notation**
   - Asymptotisches Wachstum
   - Obere und untere Schranken
   - Algorithmen analysieren



http://xkcd.com/835