Ludwig-Maximilians-Universität München München, 16.05.2019
Institut für Informatik
Prof. Dr. Thomas Seidl
Janina Sontheim, Florian Richter

Algorithmen und Datenstrukturen

SS 2019

Übungsblatt Global 2: Komplexität

Aufgabe Global 2-1 *Knobelei: Josephus Problem*

Der jüdische Historiker Flavius Josephus hielt sich 67 n.Chr. beim Kampf um die galiläische Stadt Jotapata mit 40 weiteren Männern in einer Höhle vor den Römern versteckt. Als das Versteck verraten wurde, beschlossen diese, lieber zu sterben, als den Römern in die Hände zu fallen. Da Josephus lieber das ihm angebotene freie Geleit der Römer nutzen wollte, falls er die Höhle verlassen würde, schlug er einen kollektiven Suizid vor: Alle sollten sich im Kreis aufstellen und jeder seinen linken Nachbarn töten. Das sollte wiederholt werden, bis nur noch einer übrig bleibt.

1. An welche Stelle muss sich Josephus stellen, um zu überleben?
2. Welche Position überlebt, wenn es n Personen gibt?
3. Wie stark wächst die Anzahl der Morde des Überlebenden in Abhängigkeit von n?

Lösungsvorschlag:

1.

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16 ... 41
$f(n)$	1	1	3	1	3	5	7	1	3	5	7	9	11	13	15	1 ... 19

2. Eine einzelne Person überlebt offensichtlich. Wir betrachten Runden t_i, also einen Zyklus durch alle Positionen bis die maximale Position erreicht wurde. Dabei unterscheiden wir Runden, in denen gerade oder ungerade viele Personen im Kreis stehen. Falls wir gerade viele Personen zählen, halbiert sich die Personenanzahl von einer Runde zur nächsten. Für die neue Position x gilt, dass sie vorher Position $2x - 1$ war. Beispiel: Für einen Kreis mit Positionen $1, 2, 3, 4, 5, 6, 7, 8$ entfallen zur nächsten Runde alle geraden Positionen. Für die neuen Positionen gilt die Rückabbildung $1 \to 1, 2 \to 3, 3 \to 5, 4 \to 7$. Bei ungerade großen Personenkreisen sterben $(n-1)/2$ viele Personen. Die neue Position 1 ist die ehemalige Position 3, da in t_i Position 1 Position 2 tötet und Position 1 von Position n getötet wird. Die Abbildung ist dann $1 \to 3, 2 \to 5, 3 \to 7, \ldots, (n-1)/2 \to n$. Allgemein gilt, dass die neue Position x die alte Position $2x + 1$ war. Damit können wir nun eine rekursive Funktion aufstellen, die das Problem löst:
$$f(n) = \begin{cases} 1 & , n = 1 \\ 2 \cdot f(n/2) - 1 & , n \text{ gerade} \\ 2 \cdot f((n-1)/2) + 1 & , n \text{ ungerade} \end{cases}$$

3. Die obere Schranke für die begangenen Morde durch den Überlebenden ist gegeben durch die Anzahl der Runden. Diese wiederum sind die Anzahl der Rekursionsaufrufe. Daher liefert die Rekursionsgleichung $T(n) = \max(T(n/2), T((n-1)/2)) + 1$ mit $T(1) = 0$ die Anzahl der Morde. Wir schätzen ab: $T(n) \leq T(n/2) + 1$. Mit dem Mastertheorem folgt nun $T(n) \in O(\log n)$.

Aufgabe Global 2-2 *Komplexitätsklassen*

Vergleichen Sie die Komplexitätsklassen $O(\log n)$, $O(\sqrt{n})$, $O(\log^2 n)$, $O(\log n^2)$, $O(n)$, $O(\log\log n)$ und $O(\log^k n)$ miteinander. Zeigen Sie die Korrektheit der von Ihnen gefundenen Ordnung.

Lösungsvorschlag:

Da $O(\log_2 n) = O(\log_e n) = O(\log_{10}(n)$ können wir jeweils die Basis wählen für die die Rechnung am einfachsten ist. Korrekte Ordnung: $O(\log(\log(n))) \subseteq O(\log n) = O(\log(n^2)) \subseteq O(\log^2 n) \subseteq O(\log^k n) \subseteq O(\sqrt{n}) \subseteq O(n)$

Begründung $O(\log(\log(n))) \subseteq O(\log n)$: Wir wissen $\forall n \in \mathbb{N} : \log(n) < n$.

Begründung $O(\log n) = O(\log(n^2))$: Logarithmus Rechenregeln: $\log(n^k) = k * \log(n)$. Also $O(log(n^k)) = O(k * \log(n)) = O(\log(n))$

Begründung $O(\log n) \subseteq O(\log^2 n)$: Wähle $n_0 = 2$ (und $c = 1$), dann $\forall n \geq n_0 : 1 \leq \log(n) \Rightarrow \forall n \geq n_0 : \log(n) \leq \log(n) * \log(n) = \log^2(n)$

Begründung $O(\log^2 n) \subseteq O(\log^k n)$: äquivalent.

Begründung $O(\log^k n) \subseteq O(\sqrt{n})$: Betrachte den Grenzwert

$$\lim_{n\to\infty} \frac{\ln^k(n)}{\sqrt{n}} \underset{l'Hospital}{=} \lim_{n\to\infty} \frac{k\ln^{k-1}(n) * 1/n}{\frac{1}{2}n^{-1/2}} = \lim_{n\to\infty} \frac{2k\ln^{k-1}(n)}{\sqrt{n}} \underset{l'Hospital}{=}$$
$$2k \lim_{n\to\infty} \frac{(k-1)\ln^{k-2}(n)}{\frac{1}{2}n^{-1/2}} = 2^2 * k * (k-1) * \lim_{n\to\infty} \frac{\ln^{k-2}(n)}{\sqrt{n}} = \cdots =$$
$$2^k k! \lim_{n\to\infty} \frac{ln^0(n)}{\sqrt{n}} = 0 < c < \infty$$

Begründung $O(\sqrt{n}) \subseteq O(n)$: Wähle $n_0 = 1$ und $c = 1$. Dann gilt $\forall n \geq n_0 : 1 \leq |\sqrt{n}|$. Daraus folgt $\forall n \geq n_0 : |\sqrt{n}| \leq c * |n|$. Also $O(\sqrt{n}) \subseteq O(n)$

[width=0.45]lnsqrtplot.png

Aufgabe Global 2-3 *Multivariate Komplexität*

Univariate Definition (siehe Vorlesung):

$$O(f) = \{g : \mathbb{R} \to \mathbb{R} | \exists c > 0\ \exists x_0 > 0\ \forall x \geq x_0 : |g(x)| \leq c \cdot |f(x)|\}$$

Eine mögliche multivariate Definition : Seien f und g Funktionen definiert auf einer Teilmenge von $\mathbb{R}^n$

$$O(f) = \{g : \mathbb{R}^n \to \mathbb{R} | \exists c > 0\ \exists \vec{x_0}\ mit\ \forall i \leq n\ x_i > 0\ so\ dass\ \forall \vec{x} : \forall i \leq n\ x_i \geq x_{0,i} \Rightarrow |g(\vec{x})| \leq c|f(\vec{x})|\}$$

Welche der folgenden Aussagen ist korrekt? Begründen Sie Ihre Antworten.

- $2^{(2^n)} \in O(n^{(2^n)})$

  Lösungsvorschlag:

  Wahr, da $2^{(2^n)} = 2 * 2 * 2 * ... * 2 \leq n * n * n * ... * n = n^{(2^n)}$

- Wenn $f(n) \in O(s(n))$ und $g(n) \in O(r(n))$ gilt, dann gilt auch $f(n) - g(n) \in O(s(n) - r(n))$

  Lösungsvorschlag:

  Falsch. Wähle $f(n) = 4n$, $g(n) = 3n$, $r(n) = n$, $s(n) = n$. Dann gilt: $f(n) - g(n) = 4n - 3n = n = O(n)$, aber $O(s(n) - r(n)) = O(n - n) = O(0)$

- $m^3n^2 \in O(m^2n^3)$

Lösungsvorschlag:

Falsch. Angenommen es wäre wahr. Dann würde nach Definition gelten:$\exists c > 0 \; \exists \vec{x_0} \; mit \; \forall i \leq n \; x_i > 0 \; so \; dass \; \forall \vec{x} : \forall i \leq n \; x_i \geq x_{0,i} \Rightarrow |g(\vec{x})| \leq c|f(\vec{x})|$

Wähle z.B. $\vec{x} = \begin{pmatrix} m_0 * n_0 * c \\ n_0 \end{pmatrix}$. Für dieses gilt zwar $\forall \vec{x} : \forall i \leq n \; x_i \geq x_{0,i}$, allerdings

$|g(\vec{x})| = |(m_0 * n_0 * c)^3 * n_0^2| = |m_0^3 * n_0^5 * c^3| > c * |m_0^2 * n_0^5 * c^2| = c * |f(\vec{x})|$, also $|g(\vec{x})| \nleq c * |f(\vec{x})|$

Daraus folgt $m^3n^2 \notin O(m^2n^3)$

- $log(m) * n \in O(m * n)$

Lösungsvorschlag:

Wahr. Für $\log_b(m)$ wähle $\vec{x_0} = \begin{pmatrix} b \\ 0 \end{pmatrix}$ und $c = 1$.

- $log(log(m^n)) \in O(n * log(m))$

Lösungsvorschlag:

Wahr. $p = \log(m^n) = n \cdot \log(m) \in O(n \log m)$. Außerdem gilt $\log(p) \in O(p)$. Damit auch $\log(\log(m^n)) = n \cdot \log(\log(m)) \in O(n \cdot \log(m))$

- $4mn \in O(n^3)$

Lösungsvorschlag:

Falsch. Wähle z.B. $\vec{x} = \begin{pmatrix} m_0 * n_0^2 * c \\ n_0 \end{pmatrix}$, restliche Begründung analog zu c).