



openHPI
Einführung in die Mathematik der Algorithmik
– Münzwiegen Analyse

Pascal Lenzner und Karen Seidel

Hasso-Plattner-Institut

Universität Potsdam

Aufgabe: Finde die falsche Münze mit so wenigen Wiegeschritten wie möglich!

1 Wiegeschritt = beliebig viele Münzen auf beide Waagschalen verteilen und gegeneinander wiegen



Balkenwaage



n identisch aussehende Münzen, genau eine davon ist gefälscht (leichter)

Vergleich der Ideen:

- haben bereits für 19683 Münzen verglichen

Welche Idee ist für 20000 Münzen besser?

Welche Idee ist für 100000 Münzen besser?

Welche Idee ist für 1000000 Münzen besser?

- benötigen allgemeine Analyse für n Münzen! (große n sind interessant)

Idee 1:

1 Münze wird mit allen anderen verglichen

$$n - 1$$

Idee 2:

je 2 Münzen werden entfernt

$$\left\lceil \frac{n}{2} \right\rceil + 1$$

Idee 3:

je 4 Münzen werden entfernt

$$\left\lceil \frac{n}{4} \right\rceil + 1$$

Idee 4:

je k Münzen werden entfernt, danach Idee 3

$$\leq \sqrt{n} + 1$$

Idee 5:

immer halbieren

$$\log_2(n)$$

Idee 6:

immer dritteln

$$\log_3(n)$$

Wie verhalten sich die Funktionen für wachsendes n ?

Aufgabe: Wollen Funktionen nach ihrem asymptotischen Wachstum klassifizieren

Idee 1:

1 Münze wird mit allen anderen verglichen

$$n - 1$$

$$\in \Theta(n)$$

Idee 2:

je 2 Münzen werden entfernt

$$\left\lfloor \frac{n}{2} \right\rfloor + 1$$

$$\in \Theta\left(\frac{n}{2}\right)$$

Idee 3:

je 4 Münzen werden entfernt

$$\left\lfloor \frac{n}{4} \right\rfloor + 1$$

$$\in \Theta\left(\frac{n}{4}\right)$$

$$\in \Theta(n)$$

Die Kosten der Ideen 1, 2 und 3 unterscheiden sich nur um konstante Faktoren.

Idee 4:

je k Münzen werden entfernt, danach Idee 3

$$\leq \sqrt{n} + 1$$

$$\in O(\sqrt{n})$$

Tatsächlich gilt:

$$\in \Theta(\sqrt{n})$$

Idee 5:

immer halbieren

$$\log_2(n)$$

Idee 6:

immer dritteln

$$\log_3(n)$$

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

Die Kosten der Ideen 5 und 6 unterscheiden sich nur um konstante Faktoren.

Wie verhalten sich die Funktionen $f(n) = n$ und $g(n) = \sqrt{n}$ zueinander?

Monome

$$a \leq b \Rightarrow n^a \in O(n^b)$$

$$\text{Es gilt: } g(n) = \sqrt{n} = n^{\frac{1}{2}}$$

$$\text{Es gilt: } g(n) \in O(f(n))$$

Gilt auch:
 $f(n) \in O(g(n))$?

Definition: Funktion $f(n)$ wächst höchstens so schnell wie Funktion $g(n)$, falls ab einem $n_0 \geq 0$ für alle $n \geq n_0$ gilt: $f(n) \leq c \cdot g(n)$, wobei $c > 0$ eine Konstante ist.

$$f \in O(g)$$

$$\text{Formal: } f \in O(g) \iff \exists c > 0 \quad \exists n_0 \geq 0 \quad \forall n \geq n_0 \quad f(n) \leq c \cdot g(n)$$

Gilt auch: $n \in O(\sqrt{n})$?

Antwort: Nein!

Beweis: Zeigen, dass $\neg (n \in O(\sqrt{n}))$ gilt:

$$\neg(f \in O(g)) \iff \neg(\exists c > 0 \quad \exists n_0 \geq 0 \quad \forall n \geq n_0 \quad f(n) \leq c \cdot g(n))$$

$$\neg(f \in O(g)) \iff \forall c > 0 \quad \neg(\exists n_0 \geq 0 \quad \forall n \geq n_0 \quad f(n) \leq c \cdot g(n))$$

$$\neg(f \in O(g)) \iff \forall c > 0 \quad \forall n_0 \geq 0 \quad \neg(\forall n \geq n_0 \quad f(n) \leq c \cdot g(n))$$

$$\neg(f \in O(g)) \iff \forall c > 0 \quad \forall n_0 \geq 0 \quad \exists n \geq n_0 \quad \neg(f(n) \leq c \cdot g(n))$$

$$\neg(f \in O(g)) \iff \forall c > 0 \quad \forall n_0 \geq 0 \quad \exists n \geq n_0 \quad f(n) > c \cdot g(n)$$

Definition: Funktion $f(n)$ wächst höchstens so schnell wie Funktion $g(n)$, falls ab einem $n_0 \geq 0$ für alle $n \geq n_0$ gilt: $f(n) \leq c \cdot g(n)$, wobei $c > 0$ eine Konstante ist.

$$f \in O(g)$$

$$\text{Formal: } f \in O(g) \iff \exists c > 0 \quad \exists n_0 \geq 0 \quad \forall n \geq n_0 \quad f(n) \leq c \cdot g(n)$$

Gilt auch: $n \in O(\sqrt{n})$?

Antwort: Nein!

Es gilt: $n \notin O(\sqrt{n})$.

Beweis: Zeigen, dass $\neg (n \in O(\sqrt{n}))$ gilt:

zu zeigen: $\forall c > 0 \quad \forall n_0 \geq 0 \quad \exists n \geq n_0 \quad n > c \cdot (\sqrt{n})$

Wähle: $n = \hat{c}^2 \cdot (n_0 + 2)^2$ mit $\hat{c} = \max\{c, 1\}$. Dann gilt:

$$\hat{c}^2 \cdot (n_0 + 2)^2 > c \cdot \left(\sqrt{\hat{c}^2 \cdot (n_0 + 2)^2} \right) \iff \hat{c}^2 \cdot (n_0 + 2)^2 > c \cdot \hat{c} \cdot (n_0 + 2)$$

$$\iff \hat{c} \cdot (n_0 + 2)^2 > c \cdot (n_0 + 2)$$

korrekt, da $n_0 + 2 > 1$ und $\hat{c} \geq c$

□

Aufgabe: Wollen Funktionen nach ihrem asymptotischen Wachstum klassifizieren

Idee 1:

1 Münze wird mit allen anderen verglichen

$$n - 1$$

Idee 2:

je 2 Münzen werden entfernt

$$\left\lfloor \frac{n}{2} \right\rfloor + 1$$

Idee 3:

je 4 Münzen werden entfernt

$$\left\lfloor \frac{n}{4} \right\rfloor + 1$$

$$\Theta(n)$$

bis auf konstante Faktoren gleich schnell

Idee 4:

je k Münzen werden entfernt, danach Idee 3

$$\leq \sqrt{n} + 1$$

$$\Theta(\sqrt{n})$$

Idee 5:

immer halbieren

$$\log_2(n)$$

Idee 6:

immer dritteln

$$\log_3(n)$$

$$\Theta(\log_2(n))$$

bis auf konstante Faktoren gleich schnell

wächst schnell

wächst langsam

wächst extrem langsam

Haben gezeigt:

Idee 4 ist asymptotisch echt schneller als Ideen 1, 2 und 3!

Außerdem gilt $\log_2(n) \in o(\sqrt{n})$

Ideen 5 und 6 sind asymptotisch echt schneller als Idee 4!